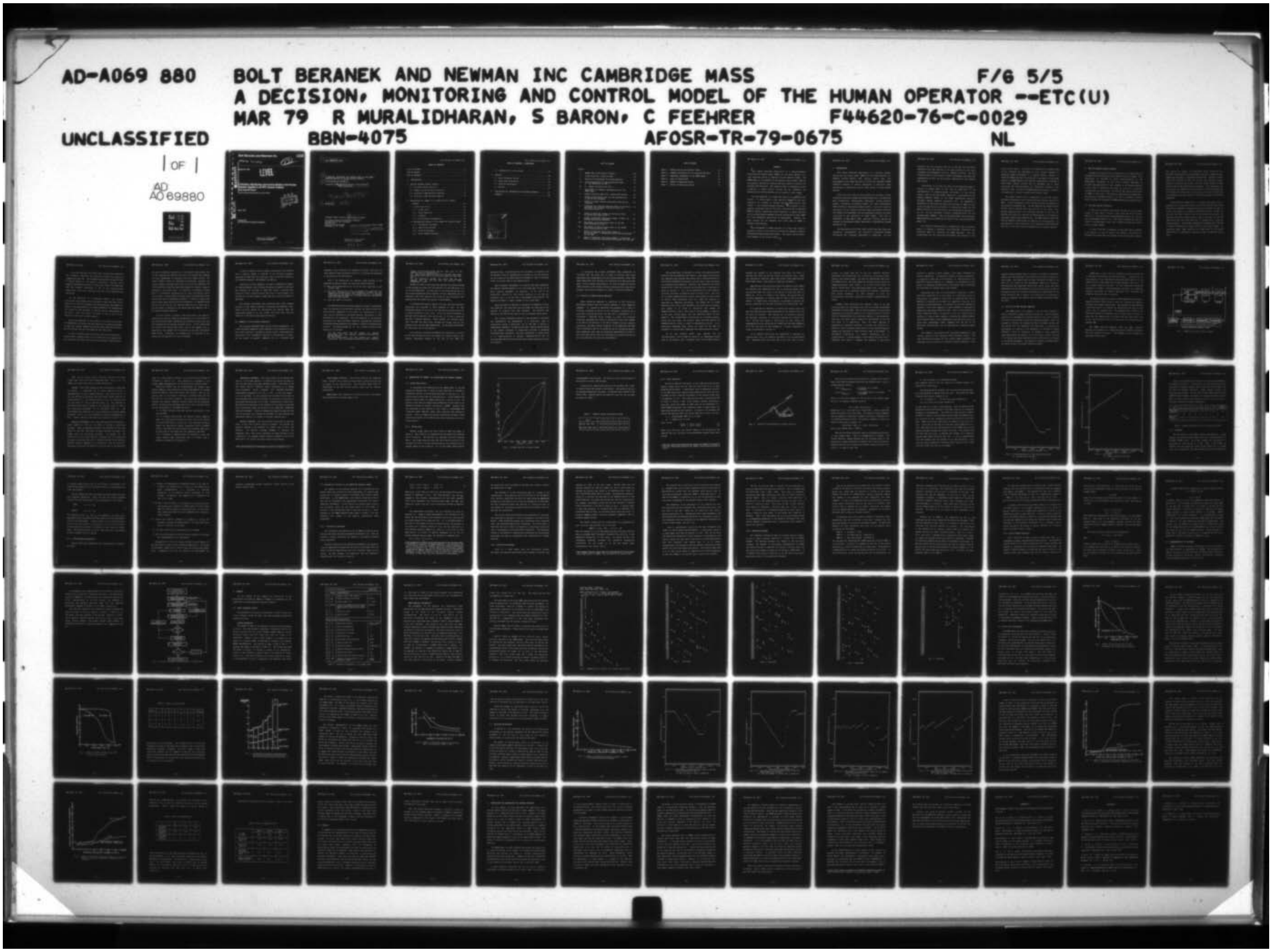
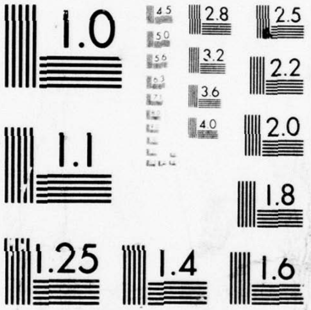


AD-A069 880 BOLT BERANEK AND NEWMAN INC CAMBRIDGE MASS F/6 5/5
A DECISION, MONITORING AND CONTROL MODEL OF THE HUMAN OPERATOR --ETC(U)
MAR 79 R MURALIDHARAN, S BARON, C FEEHRER F44620-76-C-0029
UNCLASSIFIED BBN-4075 AFOSR-TR-79-0675 NL

| OF |

AD
A069880


1.0
4.5
5.0
5.6
6.3
7.1
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

Bolt Beranek and Newman Inc.

12

Report No. 4075

A Decision, Monitoring, and Control Model of the Human Operator Applied to an RPV Control Problem

Final Scientific Report

Ramal Muralidharan, Sheldon Baron, and Carl Feehrer

March 1979

Prepared for:
Air Force Office of Scientific Research

14 BBN ~~Report No.~~ 4Ø75

6

A DECISION, MONITORING AND CONTROL MODEL OF THE HUMAN OPERATOR APPLIED TO AN RPV CONTROL PROBLEM.

9 Final Scientific Report.

Prepared by: 10 Ramal/Muralidharan, Sheldon/Baron,
and Carl/Feehrer

18 AFOSR 19 TR-79-Ø675

11 March 1979 12 96p.

Prepared under Contract No 15 F4462Ø-76-C-0029

Air Force Office of Scientific Research
Directorate of Life Sciences
Building 410
Bolling Air Force Base
Washington, DC 20332

060 100 LB

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

LIST OF FIGURES

Figure

LIST OF TABLES

ABSTRACT

This report describes application of a decision-making, monitoring and control model (DEMON) for the human operator to a task involving control of Remotely Piloted Vehicles (RPVs). The DEMON model is an extension of the Optimal Control Model (OCM) of the operator derived by infusing decision theoretic notions into the basic OCM structure. The resulting model is designed to treat situations in which control actions may be infrequent and monitoring and decision-making are the operator's main tasks.

The task modelled is a simplified version of a simulated RPV mission. It retains many of the cognitive aspects of the full simulation but differs in several details, particularly with respect to the operator/system interface. The analysis of this problem illustrates some of the major considerations in applying DEMON to complex, supervisory control problems. It shows that with fairly straightforward assumptions about the operator's task, DEMON will give reasonable predictions of performance. However, the model results are not compared with actual data so DEMON is presently unvalidated.

The development of DEMON was part of a three year research program for the Air Force Office of Scientific Research aimed at investigating human performance models. The report also provides a brief summary of the overall effort.

1. INTRODUCTION

This report describes application of a decision making, monitoring, and control model (DEMON) for the human operator to a Remotely Piloted Vehicle (RPV) enroute control task. The development of this model was part of a three year program of AFOSR sponsored research to explore approaches to modelling human performance in supervisory control tasks typical of large complex systems.

During the first year of this program, we reviewed a rather extensive literature in human performance modelling, including data bank formulations, network-based techniques, control-theoretic models, information processing models, and some miscellaneous models having an operations-research flavor. From this review we distilled a set of issues concerning human performance modelling that needed to be addressed, and we recommended research that would contribute to the resolution of those issues. This work is documented in BBN Report 3446, entitled "Critical Review and Analysis of Performance Models Applicable to Man-Machine Systems Evaluation" [1].

One conclusion of the first year's effort was that there were substantial philosophical and practical differences between "bottom-up" and "top-down" approaches to modelling. We also observed from the literature that in no case had the alternative approaches ever been applied to the same problem so that their strengths and weaknesses could be compared directly. We were interested both in the process of model development from these two perspectives as well as the relative usefulness of the products that result.

Accordingly, in the next two years of the program we began to develop an example model of each type for application to the representation of operator/system performance in the enroute control task of the RPV manned simulation (investigated by the Systems Research Branch of the Human Engineering Division, Aerospace Medical Research Laboratory [2]). We completed a formulation of the bottom-up model and delivered to AMRL the flow charts and specifications required to integrate our model into the SAINT simulation of the RPV control task developed by Wortman, et. al. [3]. We also developed a general conceptualization for the top-down model (DEMON). Both the bottom-up model and the general DEMON formulation are described in detail in our Interim Scientific Report [4].

As a result of these efforts, it became clear that it would be useful to develop a separate, self-contained, simplified RPV simulation model for analyzing the DEMON approach. This was accomplished and the resulting development and analysis is the focus of this report. Our goal is to demonstrate by example some of the strengths and weaknesses of this top-down approach to modelling human performance in a complex system. However, it should be emphasized that the structure of the model for the human operator in DEMON appears to be generally applicable to problems in which human control actions may be infrequent and in which monitoring and decision-making are significant aspects of the operator's task. As such, the DEMON approach provides a framework for modelling a variety of supervisory control tasks.

The report begins with a description of the RPV control problem as conducted in the five-station drone control facility at AMRL. A brief discussion of the bottom-up modelling approach to the problem is presented next. We then describe the simplified RPV enroute control problem which was utilized in this study. This problem captures many of the important features of the full RPV problem and allows us to develop an understanding of the modelling technique without undue complication. The DEMON model is described next and results illustrating its behavior are then presented. We conclude with a brief discussion of the modelling issues uncovered thus far in this work and with suggestions for further research.

2. THE RPV ENROUTE CONTROL PROBLEM

We begin this section with a description of an RPV mission as conducted in the five-station drone control facility at AMRL. This description is followed by brief characterization of the SAINT-based formulation of the control process developed by Pritsker and associates [3] and a summary of the general approach we have taken in modifying this model to arrive at a bottom-up representation of the process. An overview and general description of DEMON is then given. The material presented here summarizes a more detailed discussion given in [4] and is included to provide further context for the specific DEMON application discussed later.

2.1 The AMRL Manned Simulation

An RPV mission requires coordinated flights of up to eleven groups of three RPV's (triads); each group having one strike vehicle (S), one electronics countermeasures vehicle (E), and one low reconnaissance vehicle (L). The S and E vehicles fly over the target 15 seconds apart, while the L vehicle follows two minutes later to assess damage.

At launch, each RPV is assigned a flight path that is assumed to be optimal in terms of terrain and defense. The vehicle is automatically controlled with respect to this flight path; however, each vehicle is subject to flight-path errors resulting from navigation system errors, position-reporting errors, communications jamming by the enemy, or equipment malfunctions. Because of these errors and resultant drifts off course, the vehicles require external monitoring and control from the ground station to keep them as close to the desired path as possible. This supervision is provided by four human enroute controllers, who are equipped with CRT displays for monitoring of flight path and vehicle status and with keyboards and light pens for introducing changes in RPV flight parameters.

Strike RPVs are handed off to a terminal controller, who is equipped with a television picture of the view from the nose of the RPV and with standard aircraft controls and displays in order to direct each vehicle to a specific designated target, release its pay load, and hand it back to one of the enroute controllers. To simulate equivalent operations for E and L vehicles, the enroute controllers hand off these vehicles to a pseudo-pilot, using the same procedures. The operator designated as pseudo-pilot receives a vehicle by operating a toggle switch on his control panel. At specified times, these vehicles are handed back to the enroute controllers at a designated location on their pre-defined flight path.

For strike vehicles, the flight path includes three waypoints. The S waypoint identifies the position at which the vehicle is prepared for hand off. The H waypoint designates the desired point of actual handoff to the terminal controller. Finally, the B waypoint designates the point at which the vehicle is handed back from the terminal controller to one of the enroute-return controllers. For E and L vehicles, only the H and B waypoints are identified.

At the beginning of a simulated mission, the enroute controllers first examine the pre-scheduled times that each strike vehicle is to arrive at handoff; they then generate, with paper and pencil, a revised schedule that spaces handoffs to be separated by two minutes so that overlaps in terminal control requirements do not occur. They also adjust the speed of one or more strike vehicles to meet this revised schedule.

During the remainder of the mission, the enroute controllers are responsible for monitoring the flight path of vehicles assigned to them, for issuing commands correcting flight path and velocity, and for dealing with any contingencies that may arise.

In order to conduct these activities, they are provided with a listed/tabular summary status for all RPVs and with capabilities for displaying the flight path and detailed status of each vehicle.

The entire simulation operates on a five-second frame update rate, so that displays are updated once each five seconds and commands are only implemented in synchrony with this update period. The status summary, which is displayed continuously, presents the vehicle number, estimated time of arrival at the next waypoint, and a three-character code that describes command link status, waypoint designator, and flight mode. In addition, it displays a number which is incremented automatically for each five-second period during which a given vehicle deviates from the prescribed flight path by more than an adjustment threshold. In order to examine the actual flight path, detailed vehicle parameters, or commands issued but not yet carried out, the operator must point his light pen at the RPV number in question on the status menu and depress a key on the special-purpose keyboard.

To enter a patch (a change in RPV flight path), the operator indicates the desired change by designating one or more points on the revised flight path, depressing the reconnect function key, and then designating the desired reconnect point. If the change does not violate turn-radius constraints, and if the command link is operational, the command will be executed at the next five-second frame update. Otherwise, the command will be rejected by the system and the operator will be so informed.

To enter a change in vehicle speed, the operator must indicate that a velocity change is required on the function keyboard, designate the RPV with the light pen, type in the new velocity on the standing keyboard, and depress the EOB key.

Just prior to the S waypoint, an S RPV is prepared for handoff by a pop-up maneuver that includes changing its speed to 250 knots and changing its altitude to 3000 feet using a function key. Pop-up for E and L vehicles occurs just prior to the H waypoint and involves an altitude change to 3000 feet and a velocity change to 400 knots.

The enroute controllers are instructed that their highest priority is the timely execution of pop-ups and handoffs, their second priority should be maintaining the desired ETAs and separations between S, E, and L RPVs, and their third priority should be to minimize flight path deviations.

2.2 Summary of the Pritsker Simulation

The original SAINT/RPV model has two primary components: (1) a state variable component, which consists of the simulation of the RPV flight position, navigation system errors, maneuverability constraints, fuel consumption, effects of disturbances on flight, and the impact of operator commands; and (2) a discrete task component, which simulates the sequence of control, decision, and other operator tasks reviewed in Section 2.1 that must be performed in carrying out the RPV mission.

With a few exceptions, all operator tasks defined in the SAINT/RPV simulation share the following characteristics:

(1) They can be performed by any one of the four operators on the control team.

(2) The time required for their performance are selected from specified distributions, most frequently normal, and are rounded off to the nearest five-second interval. All elapsed times employed are equal to or greater than zero seconds and less than 9,999 seconds.

(3) They are equal in priority.

The SAINT/RPV simulation model embodies a number of mechanisms that are required for coordination between the state-variable and task-oriented components of the model, for computation of task time, and for matching of simulated operator performance to that exhibited by real operators. One such mechanism is the Operator Attribute file, which provides a means for representing individual differences among operators with respect to decision thresholds and criteria. Following Wortman et. al. [3], a short catalog of such factors is as follows:

> "(1) The time before the RPV reaches its handoff coordinates that the operator prefers to initiate the pop-up maneuver;
> (2) the times before the RPV reaches its handoff coordinates that the operator prefers to make a velocity

> change, the altitude change, and the hand over to the terminal pilot or pseudo-pilot;
> (3) The lateral deviation value for the RPV above which the operator will make a directional change for that RPV; and
> (4) The difference between the actual ETA and the desired ETA of the RPV that the operator deems acceptable." (p.40)

Values for each of the twenty-two Operator Attributes defined in the program are input for each operator on the team prior to a run of the simulation. As each task is initiated during the run, the program determines which operator will be responsible for its execution, and then acquires the values of the attributes that characterize the identified operator's performance.

A simulated RPV mission begins with each simulated operator monitoring the progress of RPVs assigned to him. He then determines whether or not one of the vehicles has reached the point at which he prefers to pop it up. If so, the pop-up procedure is executed and the operator then waits until it is time to hand the RPV off to another operator. After handoff, the operator waits until the RPV has been flown through the target area by the terminal pilot and has been handed back. He then pops the RPV down for the return leg of the mission.

If no pop-up or pop-down procedures are called for, as would be the case during early and late stages of a typical mission, the operator determines whether or not any of his RPVs are malfunctioning. Any malfunctions are corrected, if possible, and the operator turns to consideration of whether or not the velocity of one or more of his RPVs should be changed in order to minimize errors in arrival time at the handoff point.

When necessary adjustments in velocity have been completed, the operator decides whether or not the flight path of any of his RPV's requires amendment (or patching). If so, and if all constraints relative to the current position of the RPV are satisfied (e.g., it is not near a programmed turning point), the operator proceeds to input a change in the flight path.

Returning RPVs are checked to determine the adequacy of their fuel supplies, and velocities and altitudes are modified by the operator to conserve fuel when necessary. The operator then returns to the monitoring function and the process begins again.

The original SAINT/RPV simulation was designed to replicate the organization and performance of a particular team of controllers during a particular run of the RPV II series of experimental missions. To achieve this goal, several modifications to the general character of operations outlined in preceding paragraphs were introduced. The most significant of these relate to (1) specialization of operator responsibilities and (2) pre-programed hand-off failures and other missed operations during the mission.

In evaluating the original SAINT/RPV model components, we concluded that the areas most in need of revision were those associated with system monitoring and decision-making, rather than those associated with carrying out decisions once they have been made. As will become clear shortly, we chose to merge the various tasks involved in the primary decision and monitoring loop and to preserve the existing integrity of tasks in decision execution.

2.3 Overview of BBN Bottom-Up Approach

BBN's bottom-up approach to modelling of RPV controller performance differs from the original approach in three important respects: (1) instead of searching one parameter at a time, it utilizes a paradigm in which all the information available for a given RPV is extracted before the next RPV is considered; (2) it introduces a deferred action concept in which the simulated controller postpones the taking of corrective action with respect to an RPV of low priority if an RPV of higher priority requires correction, and then returns attention to the deferred item when time is available; and (3) it avoids the use of "regression models" with parameters that must be determined experimentally within a particular application, and utilizes models with greater generality for the prediction of controller performance.

The revised model is designed to reflect the complex priority structure that the operators must employ. Some types of deviations are inherently more serious than others, but a large deviation on a low-priority dimension can be more critical than a small deviation on a high-priority dimension. Moreover, the importance of a given deviation will often be a function of how much time is available in which to correct it. As a first approximation to this priority structure, the model includes two sets of action limits. The first set is termed immediate-action limits, and consists of those values of various state deviations that will cause the operator to institute an immediate correction. The second set is termed deferred-action limits, and represents the values that the operator will employ if he finds no deviations that exceed the immediate-action limits. Both sets of limits depend, in general, on RPV type, mission phase, and time remaining before the next waypoint. The revised model is structured as a two-pass process. During the first pass, the operator checks each RPV against the immediate-action limits in order of descending priority. If no deviations exceeding these limits are found for any RPV, he proceeds to a second pass, employing the deferred-action limits.

As in the original model, each operator has an "enroute/return" list of RPVs for which he is responsible during most of the mission, and a "terminal area" list of RPVs which he prepares for handoff to the terminal area pilot and which he receives back when the RPV has cleared the terminal area. These two lists may be identical under some control team organizations, while under others, they might be completely different.

Upon entering the monitoring phase the operator first checks his terminal area responsibility list to determine whether there are any RPVs that are close to the points at which they must be popped up. If so, he checks to see whether the pop-up must be initiated immediately or whether there is time to carry out other checks. If insufficient time remains, he proceeds to perform the pop-up; otherwise, he continues checking his terminal-area list.

If no pop-ups are imminent, he checks his terminal-area list again for RPVs that have been handed back by the terminal pilot and are ready for pop-down. Upon finding one, he proceeds to perform the pop-down; otherwise, he continues checking his terminal area list. During this phase of monitoring, he also checks for unacceptable lateral deviations for S RPVs that have been popped up at S, but that have not yet reached H. If such a deviation is found, he proceeds to correct it.

If no required activities are discovered in checking the terminal-area list, the operator proceeds to his enroute/return list. Beginning with the first RPV on his list that is still enroute, he checks each RPV in turn for required reroutes, reprograms, and malfunctions (all of which are initiated by other tasks of the original model), and then for unacceptable ETA errors and LATDEV errors. He first checks for serious errors, using the "immediate-action" error limits. If no such errors exist, he begins a second pass. This pass begins again with a check of his terminal-area list to determine whether any pop-ups or pop-downs have become necessary. He then proceeds to check his enroute/return list again using more stringent, "deferred-action" limits.

Elapsed times associated with operator tasks in the BBN bottom-up model are calculated with the aid of human performance sub-models and algorithms. Each of these sub-models and algorithms represents, with as much fidelity as is possible given our current state of understanding, the structural aspects of the perceptual, cognitive, and motor skills required in performance of the task with which it is identified. Thus, the model employed in simulating an operator's decision to correct the velocity of a vehicle in order to assure timely arrival at the hand-off waypoint assumes the existence of three distinct types of processes: (1) information acquisition, (2) numeric estimation, and (3) classification. A second example is provided by the model used in computing time taken to complete the sequence of operations involved in issuing a patch command. This model envisages six distinct operations: (1) scanning a display to obtain information, (2) pointing a light pen at the display, (3) identification of a particular function button on a keyboard, (4) depression of the button, (5) scanning the RPV track on the display, and (6) pointing the light pen at a second area of the display.

Associated with each of the perceptual, cognitive, or motor operations identified in a given model is a particular value or distribution of completion time and/or an algorithm that can be employed to generate an estimate of completion time for that operation during simulation. An estimate of the time required to complete a total task composed of these elements is achieved by summing the individual operation times. Thus, in the second of the models summarized above, an estimate of the time required by an operator to issue a patch command is achieved by adding together three scanning times, drawn independently from one distribution, to three motor performance times, computed with the aid of two additional distributions, and an algorithm for combining sample values.

The temporal distributions and algorithms employed in the current version of the simulation have different origins. One type, specified by Pritsker for the original SAINT simulation, was developed from the results of the RPV II system simulation. This category contains models that are similar in concept to regression models, and that "describe" very accurately the results obtained in that study. The second type has been introduced by BBN on the basis of its review of the performance modelling literature. Distributions and algorithms having this latter origin have been substituted for the corresponding Pritsker formulations as part of the general effort to increase the generality of application of the SAINT simulation and to explore the feasibility of developing a bottom-up approach that employs existing human performance models and data.

2.4 Overview of BBN Top-Down Approach

The DEMON model is an example of the so-called top-down or analytic approach to human performance modelling. Such an approach begins with a mathematical characterization of the task including the overall goals and the criteria for good performance. Then, one attempts to develop the assumptions about the human operator and the system that are necessary and sufficient to characterize performance in relation to the parameters of interest to system designers. There is, generally, an attempt to avoid modelling human performance and behavior at a micro-level; rather, the hope is to capture just those aspects that are significant with respect to the design parameters. Two important classes of top-down models are manual control and signal detection models.

DEMON does, in fact, have its foundation in control theory and in statistical estimation and decision theory. In particular, it draws heavily on the information processing model implicit in the optimal control (OCM) model of the human operator (see, e.g., [5] for a recent review of the OCM). To this information processing structure is added a decision-making structure for modelling discrete monitoring and control decisions and a structure for computing continuous control actions.

The decision making structure in DEMON embodies the concept of expected net gain (ENG), which is used as a criterion for making a rational choice among alternatives. The expected net gain ENG from a particular action is obtained by subtracting the cost of that action from its expected gain. The expected gain itself is the difference between the expected cost of events when no action is taken and the expected cost of events that may arise after this action. The rational choice is to select that action which has the greatest ENG.

The DEMON modelling approach views the human (enroute) operator as an element in a closed-loop control system, as shown in the block diagram of Figure 1. The elements in this diagram are described briefly below.

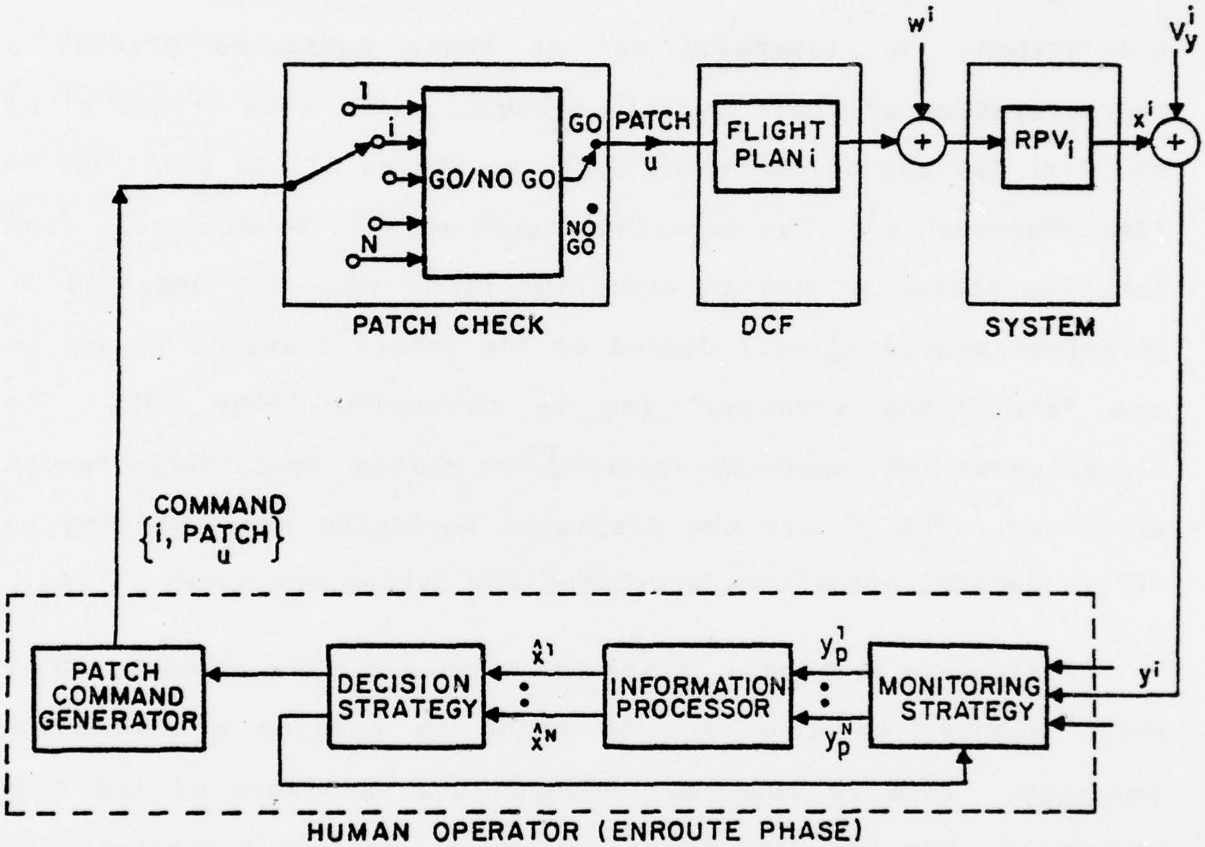


Fig. 1 DEMON View of RPV Control Problem

DCF: The DCF (Drone control facility) contains the stored flight plans that drive the N subsystems RPV_i, $i=1,2,\ldots,N$. For DEMON, the flight plans may be chosen arbitrarily.

System: The N RPVs undergoing monitoring/control constitute the system. A linearized set of state equations provide a representation of their dynamic behavior. The true status x^i of the i-th RPV may be different from the stored flight plans due to "disturbances" w^i. The reported status y^i will be different from the true status x^i due to reporting error v_y^i. The observed or perceived status y_p^i will depend on the reported status y^i and on the "monitoring strategy" (to be discussed later on). The disturbances and reporting error v_y^i are modeled by suitable random processes. The y^i are the displayed variables corresponding to RPV_i. General equations describing the system are given in [4].

Monitoring Strategy: Since the human must decide which RPV or which display to look at, he needs to develop a monitoring strategy. This is important because his estimates of the true status of each RPV (and hence his patch decision strategy) will depend upon his monitoring strategy. To account for the interaction of the patch decision strategy with the monitoring strategy we formulate and solve a combined monitoring and patching decision problem (Appendix B of Reference 4 has the details). Separating the monitoring decisions from the rest of the decisions leads to a much simpler derivation of the monitoring strategy as discussed in Section 3.2. Such separation is assumed in the present DEMON implementation of the RPV enroute operator model.

Monitoring models may be distinguished by whether they predict temporal (time histories of) monitoring behavior or average monitoring behavior over some chosen time horizon. Most of the earlier work in the control literature, including that with the OCM, falls in the latter category. The monitoring strategies we consider will predict temporal behavior which can be simulated. Some of the monitoring strategies derived in the literature which can be investigated in the DEMON setup are:

(i) A simple strategy involving cyclical processing of the various RPVs.

(ii) A strategy generalizing the Queueing Theory Sampling Model [6], which would minimize the total cost of not looking at a particular RPV at a given time. This strategy is mainly useful for maintaining lateral deviations within allowable limits. The costs for errors and for the different RPVs would be functions of the time-to-go and, possibly, RPV type.

(iii) A strategy of sampling when the probability that the signal exceeds some prescribed limit is greater than a subjective probability threshold [7].

Information Processor: This block models the processing that goes on in the human operator to produce the current estimate of the true RPV status from past observed status. This block is the well known control- theoretic model consisting of a Kalman filter-predictor which produces the maximum-likelihood, least-squares estimate $\hat{x} = (\hat{x}^1, \hat{x}^2, \ldots, \hat{x}^N)$ of the true status x of all the RPVs. It also produces the variance of the error in that estimate.(Note that an estimate of the state of each RPV is maintained synchronously at all times. Observation of a particular RPV improves the accuracy of the estimate of the status of that RPV while uncertainty about the status of the remaining, unobserved vehicles increases.) Given the assumptions generally made for this kind of analysis, the information processor can thus generate the conditional density of x based on the past observations y.

Decision Strategy: This block models the process of deciding which, if any, RPV to patch, pop-up or handoff. We consider the decision process to be discrete (it takes 5 sec to get a new display). The decision strategy attempts to maximize the (expected) gain.* This block translates the best estimate $\hat{x}$ into a decision to (i) command a patch, pop-up or handoff to one of the RPVs and/or (ii) modify the future monitoring strategy.

* The cost functions are discussed in detail in Section 3.2.

Patch Command Generator: This block generates the commanded patch. We shall use a simple strategy based on minimizing the time to return to the desired path. The allowable paths would be constrained by the RPV turning radius limits and the velocity constraints.

Patch Check: This consists of a GO/NO GO check on the patch using conditions on turning radius, etc.

3. APPLICATION OF 'DEMON' TO A SIMPLIFIED RPV CONTROL PROBLEM

3.1 System Description

In developing and demonstrating the DEMON model in the RPV context we decided that it would be most efficient to consider a simplified version of the problem described in Section 2.1. The version used in this study is discussed below. A more complete and general system model for DEMON application is given in [4]. We hasten to note that the simplifying assumptions we have made do not generally reflect an inherent limitation in DEMON. Instead, they are motivated by our desire at this stage to investigate and illustrate model behavior rather than replicate RPV simulation results. With our simplified problem formulation we hope to have captured the essence of the RPV mission while discarding the nitty gritty details.

3.1.1 Flight Plan

Nominal flight plans for each triad of RPVs are shown in Figure 2. The S-vehicle is launched first followed by an E-vehicle and an L-vehicle. The vehicles are launched forty-five seconds apart. The flight plans are such that the E-vehicle arrives in the terminal area first, fifteen seconds ahead of the S-vehicle and 9Ø seconds ahead of the L-vehicle. All of the flight plans contain

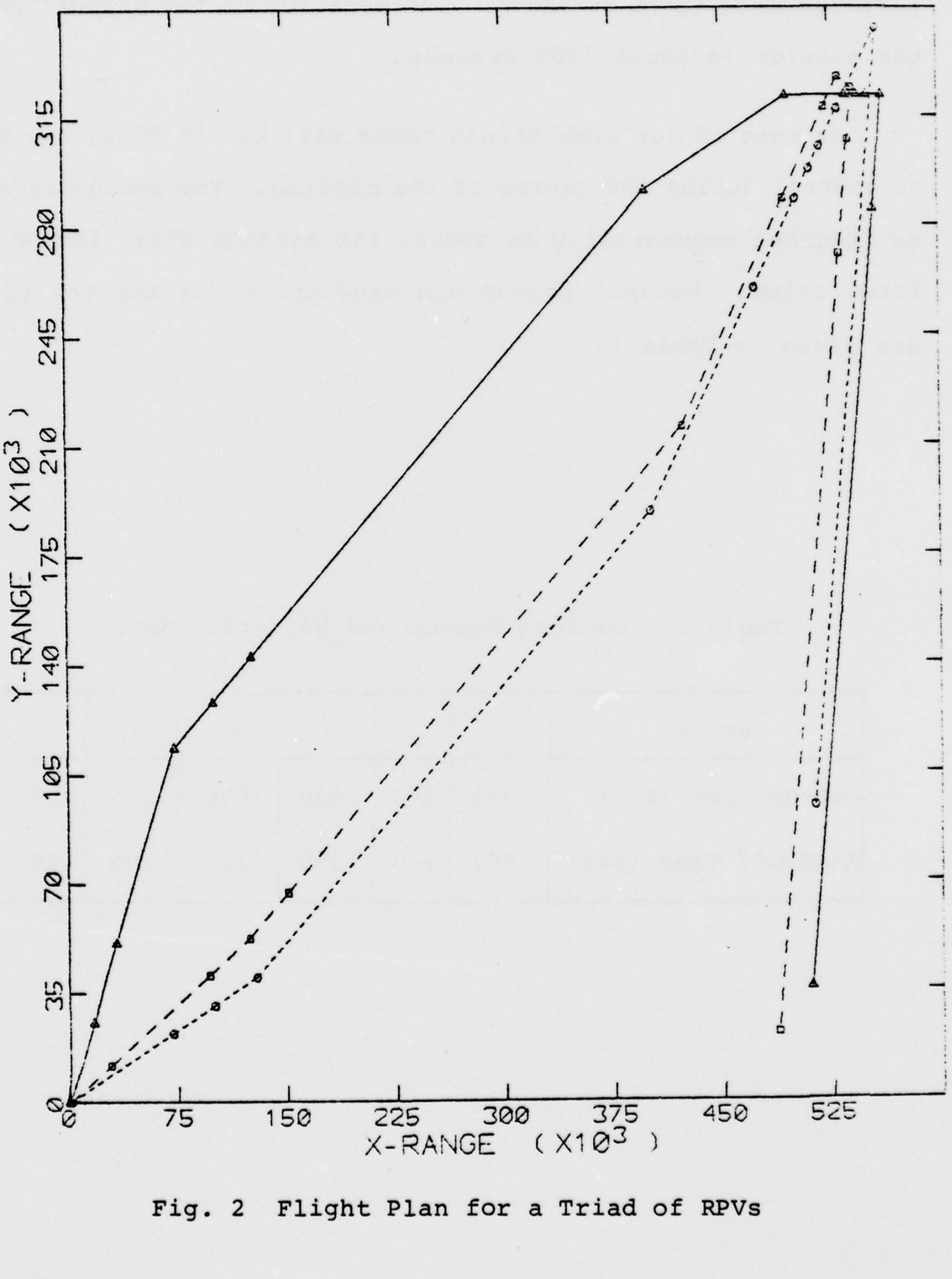


Fig. 2 Flight Plan for a Triad of RPVs

preprogrammed turns as shown. The duration of the enroute phase of the mission is about 1200 seconds.

In most of our simulations there will be six RPVs, two triads, to control during the course of the mission. The second triad will be launched sequentially as above, 135 seconds after launch of the first triad. Nominal pop-up and hand-off times for the six RPVs are given in Table 1.

Table 1 Nominal Pop-up and Hand-off Times

RPV #	1	2	3	4	5	6
Pop-Up Time (sec)	955	940	1030	1090	1075	1170
Hand-Off Time (sec)	980	965	1055	1115	1100	1190

3.1.2 State Equations

We wish to describe the manner in which RPVs deviate from the nominal flight plans and how they are controlled so as to keep these deviations small. A description that is completely faithful to the AMRL simulation requires state equations for error and rate of change of error along and orthogonal to the nominal flight path (see Appendix A, [4]). This requires four state variables per RPV as shown in Figure 3 and, given present program constraints, seriously limits the number of RPVs that can be considered in a single DEMON simulation.* We, therefore, chose a simpler model for describing deviations from nominal. In Figure 3 let B denote the desired RPV position and P the actual position at time t so that

x_1 = ground speed (ETA) error in feet

x_2 = LATDEV error in feet

Then, we set

$$\dot{x}_1(t) = u_1(t) + w_1(t) \qquad (1)$$

$$\dot{x}_2(t) = u_2(t) + w_2(t) \qquad (2)$$

where $u_1(t)$ and $u_2(t)$ are control inputs to be selected by the operator and $w_1(t)$ and $w_2(t)$ are disturbances causing flight plan errors.

* One can, after some manipulation reduce the number of states to three by considering equations for speed, cross-track and heading errors.

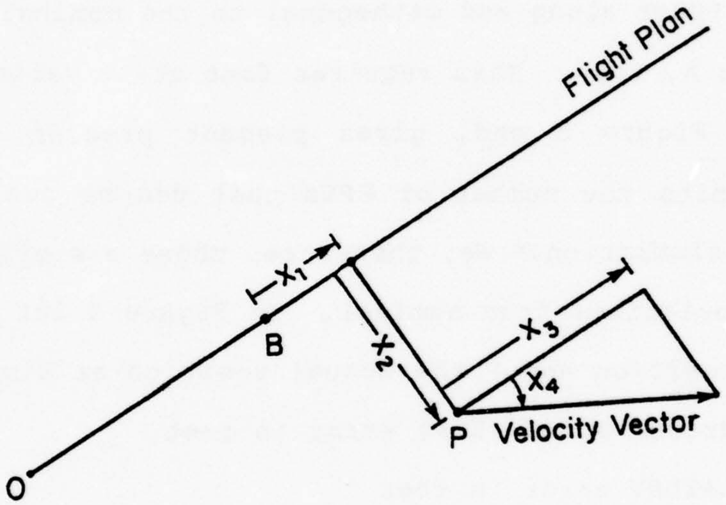


Fig. 3 Choice of Co-ordinates for System Equation

The control variables can be defined in a manner which retains some of the coupling between ETA errors and LATDEV errors. This is accomplished by letting

$$\begin{aligned} u_1(t) &= 0 && \text{if no patch is in effect} \\ &= V\cos\phi(t) - \bar{V} && \text{otherwise} \\ u_2(t) &= 0 && \text{if no LATDEV patch is in effect} \\ &= V\sin\phi(t) && \text{otherwise} \end{aligned} \tag{3}$$

where V is the true airspeed, $\bar{V}$ = 675 ft/sec is the nominal speed and $\phi(t)$ is a pseudo "heading" given by

$$\phi(t) = \tan^{-1} (x_2(t)/x_1(t)) \tag{4}$$

Equations (1)-(3) are shown for a single RPV. Similar equations govern the flight of the other RPVs. The operator's patch commands then correspond to selecting a change in velocity or heading so as to eliminate perceived errors. The possible changes are limited by vehicle constraints on speed, viz.

$$420 \text{ ft/sec} = V_{min} < V < V_{max} = 800 \text{ ft/sec} \tag{5}$$

and on turn radius, Rmin = 5280 ft.

The turning radius constraint must be introduced artificially in this simple model. This is accomplished as follows. The current position, pseudo heading and the reconnect point on the flight plan uniquely determine a "pseudo" turn circle of radius R. Assuming that the turn is made with constant velocity V over a turn time T it is easy to show that

$$R = VT/\text{turn angle} = 0.5\ VT/\text{heading change}$$

This computed value of the turn radius R is checked against the minimum turn radius R_{min}.

The disturbances are constant over any leg of the flight plan, but a new constant is chosen after each turn. The values for these constants were chosen randomly to be

$$\begin{aligned} w_1 &= -6 \text{ ft/sec or } + .3 \text{ ft/sec with equal probability} \\ w_2 &= \pm.8 \text{ ft/sec with equal probability} \end{aligned} \qquad (6)$$

in each leg for our basic investigations. Once the pseudo-random sequence of errors was selected it was kept fixed for all remaining parametric investigations. The effect of these errors is shown, for RPV1, in Figure 4 which presents ground speed and LATDEV errors as a function of time, assuming no patch corrections to the flight path. These curves also demonstrate an anomaly in the error at a turn. Given that an RPV does not have zero ground speed error at a turn, then there will be a discontinuity in the LATDEV error as the mission proceeds through the nominal turn time. This results from the fact that this error is defined with respect to a different nominal path before and after the turn. It is our understanding that this anomaly was also present in the RPV simulation. This could be easily remedied by avoiding sharp corners in the flight plan but we have not done so at present.

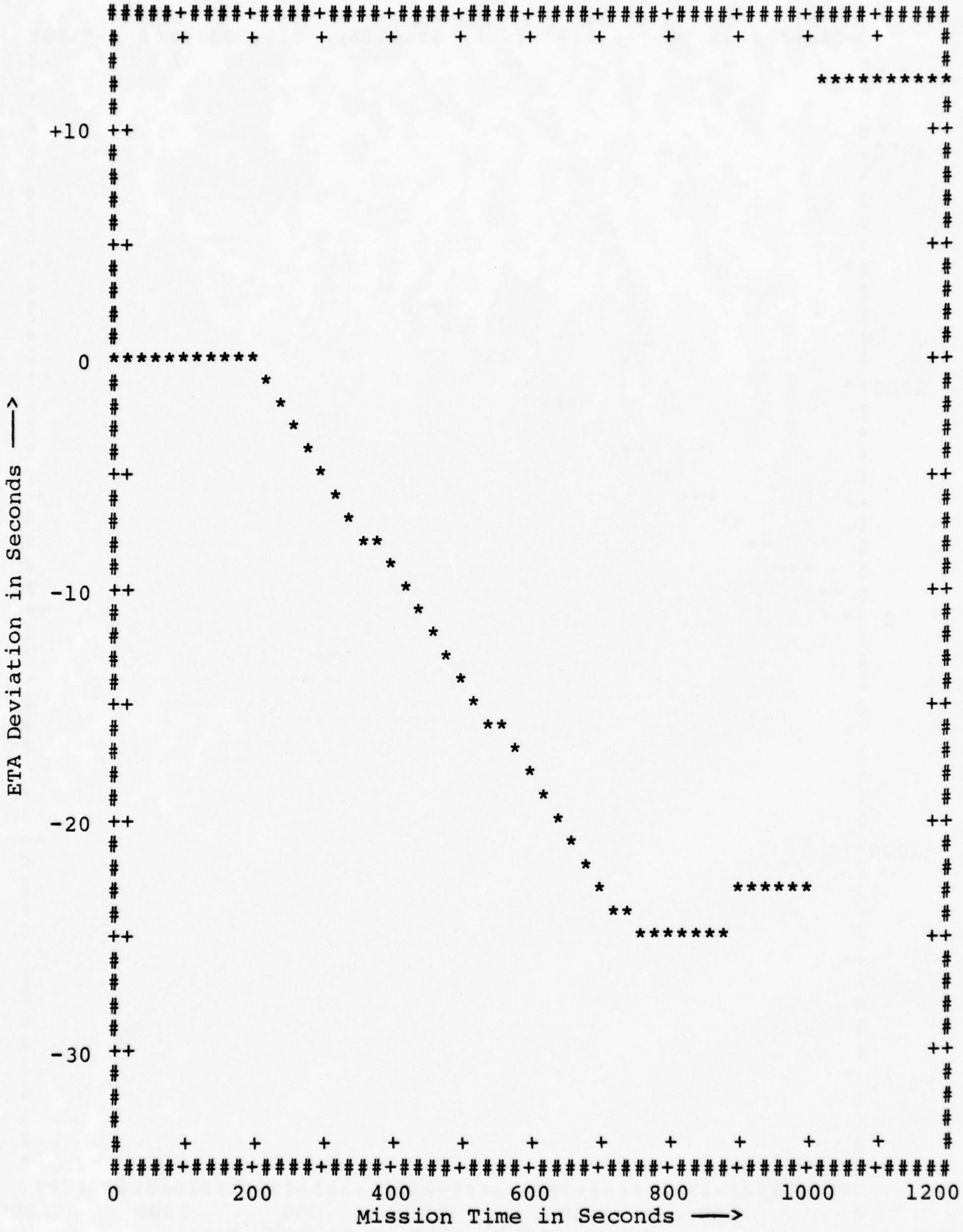


Fig. 4 State Trajectories of the Uncontrolled RPVs

a) ETA Deviation for RPV 1

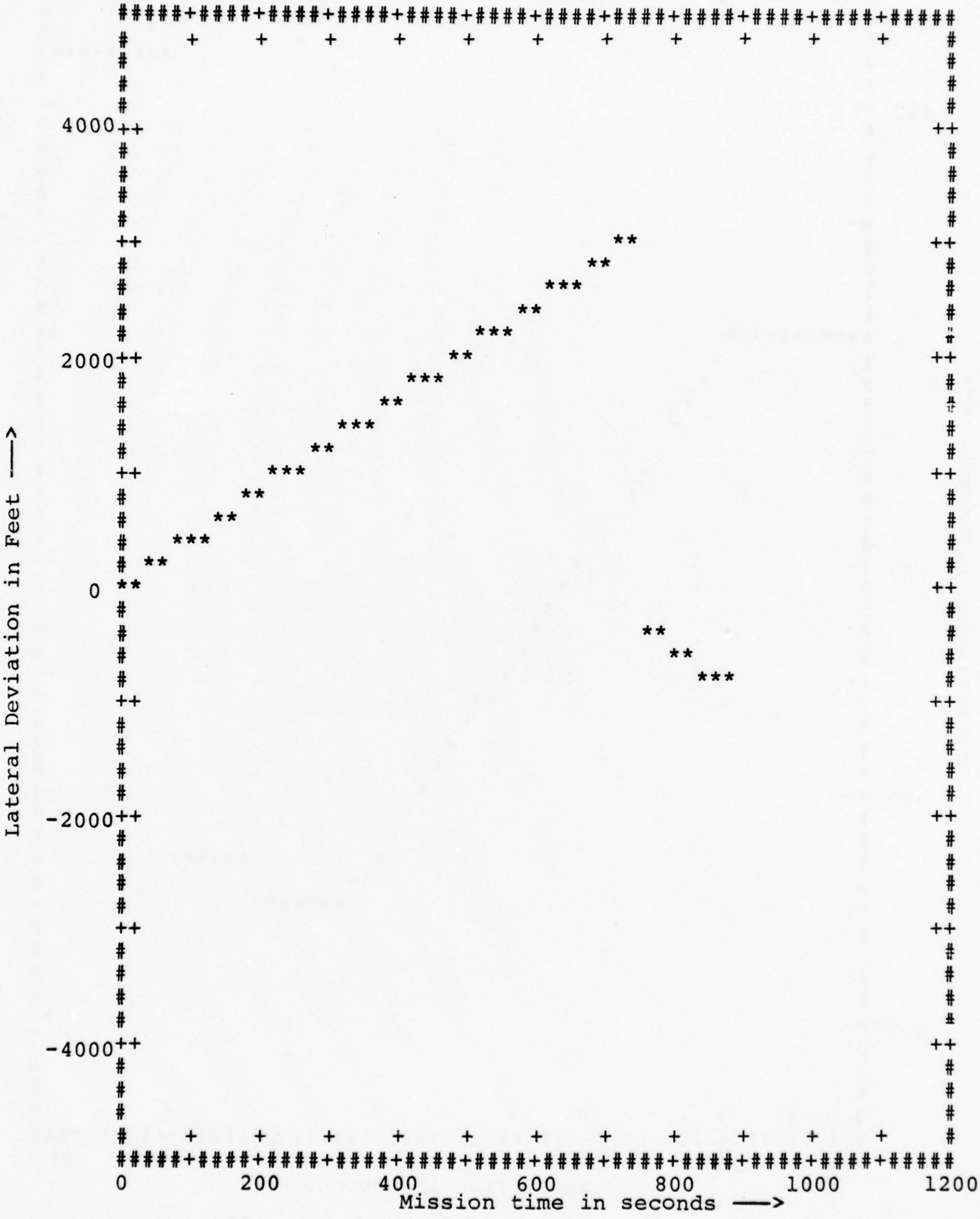


Fig. 4 Continued

b) Lateral Deviation for RPV 1

Table 2 gives summary statistics for the six simulated RPVs in the absence of control. In particular, we list the rms errors and the errors at nominal handoff time for each RPV and the mean value per RPV and standard deviation. These numbers indicate the difficulty of the task. We note that all RPVs exceed the tolerable LATDEV errors of 15ØØ ft. Moreover, if uncorrected, ETA errors for RPVs 1,2,4 and 6 would also exceed the 15 sec (1.Ø125 x $1Ø^4$ ft.) tolerance required for proper sequencing.

RPV #	1 (S)	2 (E)	3 (L)	4 (S)	5 (E)	6 (L)	Mean	Standard Deviation
RMS Ground Speed Error (1000 ft)	10.54	13.56	8.26	14.74	8.47	12.72	12.10	2.80
RMS LATDEV Error (1000 ft)	6.06	7.10	5.10	6.75	4.61	4.46	3.45	1.15

Table 2 Summary Statistics for Six Uncontrolled RPVs

3.1.3 Displays

We will assume a very simple display configuration. A menu display (M) provides ground speed error and a status display (S) provides LATDEV error. On either display the information is available for only one RPV at a time. Thus, in any frame, the operator can only determine either ground speed error or LATDEV error for a single RPV. The RPV simulation has a more complex set of display possibilities but at this stage of development this simplifying assumption will help significantly our understanding of model performance.

We also assume that there are reporting errors associated with the displayed quantities. Thus, the menu and status displays provide, for the selected RPV, noisy information on the states:

$$\begin{aligned} &\text{MENU:} && y_1 = x_1 + v_{y_1} \\ &\text{STATUS:} && y_2 = x_2 + v_{y_2} \end{aligned} \tag{7}$$

The reporting errors v_{y_1} and v_{y_2} are assumed to be white noise sequences with an autocovariance that scales with the mean-squared level of the measurement. A purely additive noise could have been used but this was simpler to incorporate via our existing program and the scaling noise model is frequently a more realistic sensor model. The signal-to-noise- ratio for reporting errors was assumed to have a nominal value of -20 dB.

3.1.4 Miscellaneous Assumptions

Several additional assumptions were incorporated to simplify the model:

i) Patches are implemented as commanded except in the case of a NO/GO. No consideration is given to the mechanics of patch implementation, i.e. key-board or light-pen operations, or to possible errors introduced in this process. In addition, the command link is assumed to be "up" at all times.

ii) To make a velocity patch it is necessary to be monitoring the menu display and, likewise, a LATDEV patch can only be made when the status display is being monitored. On the other hand, pop-up and hand-off can be commanded while looking at either display.

iii) Pop-up and hand-off commands are assumed to include the necessary altitude and speed patches. No "dead time" after pop-up or hand-off is included.

iv) No fuel constraints or malfunctions are included.

v) Only the enroute part of the RPV control problem is included and reprogramming is not considered.

Assumptions (i)-(iii) are largely related to motor aspects of the task which we view to be of secondary importance at this stage of development (especially in light of the five second frame update rate). Assumptions (iv) and (v) simplify the development and analysis considerably without affecting a major portion of the enroute control task.

3.2 Mathematical Details of the DEMON RPV Operator Model

The essence of the top-down approach is to characterize the mission goals and criteria for good performance in a manner that allows one to predict operator strategies and overall system performance. In the DEMON model, this implies selecting criteria that can be translated into appropriate monitoring and control (patching) strategies. In this section we discuss these criteria and strategies, which are the prime concern of this study, in relation to the problem described in Sections 2.1 and 3.1. Other elements of the DEMON operator model shown in Figure 1 are also discussed with respect to the specific application being considered.

3.2.1 Information Processor

The information processing portion of DEMON is drawn directly from the OCM which has been documented extensively [5]. Here, we discuss it briefly confining our remarks to the specific problem being considered.

The basic function of the information processor is to generate a set of expectations concerning the status of the RPVs, that are based on previous observations and on an "internal" model for the system. We shall assume, for simplicity, that the operator's internal model for the RPV is given by

$$\begin{aligned} \dot{x}_1(t) &= u_1(t) + w_{m_1}(t) \quad ; \quad x_1(\emptyset) = \emptyset \\ \dot{x}_2(t) &= u_2(t) + w_{m_2}(t) \quad ; \quad x_2(\emptyset) = \emptyset \end{aligned} \qquad (8)$$

where $x_1(t)$, $x_2(t)$ and $u_1(t)$ and $u_2(t)$ are the states and controls, defined in equations (1)-(3). The "disturbances", w_{m_1} and w_{m_2}, assumed in this internal representation are zero mean, gaussian white noise sequences with autocovariances $W\emptyset M_1$ and $W\emptyset M_2$ and, therefore, differ from the "true" disturbances defined by Equation (6).

The measurements available from the displays are given by equation (7). Based on these measurements, the estimates for the states and the variances of the estimation errors are given by the well-known Kalman filter equations [8]. It is straightforward to show that in the absence of control and monitoring, the estimate of the state, $\hat{x}(t)$, remains constant at $\hat{x}(t_o)$ and the uncertainty in that estimate grows as $W\emptyset M*(t-t_o)$, where t_o is the time of last observation.* If control is applied subsequent to t_o, but no further observations are made, the estimate is computed from

$$\hat{x}(t) = \hat{x}(t_o) + \int_{t_o}^{t} u(t)\, dt \ .$$

* The assumption of white noise disturbances in the internal model of Equation (8) prevents the DEMON "operator" from estimating the rate at which errors are changing due to the biases introduced by the w_i's of equation (6). Actual operators are probably capable of performing such estimation and it would be easy to include this capability in DEMON. However, to do so requires increasing the dimension of the state and, as noted earlier, this would reduce the number of RPVs that could be considered simultaneously.

The uncertainty still increases at the same rate, however, unless a new observation is made.

The estimate $\hat{x}$ is the conditional mean of x based on the observations. This quantity and the variance of the estimation error provide a sufficient statistic for specifying the **subjective** probability distribution of x, given the assumptions we have made. The above discussion shows that patching (u) affects the mean of the subjective distribution whereas monitoring affects both the mean and the uncertainty.

The initial state and uncertainty are parameters of the model that describe the operator's initial knowledge of the state of the system. These parameters become less significant with increasing time. The covariance WØM is also a parameter. As we have seen, it describes the rate at which the operator's uncertainty grows with time in the absence of any additional information. This parameter relates to the operator's expectations concerning the disturbances perturbing the path as determined from instructions or through training.

3.2.2 Monitoring Strategy

Prior to a frame update when new information becomes available, the operator must decide which display to monitor (and perhaps act upon) in the next frame. Because there are two separate displays for ETA and LATDEV for each of the N RPVs and because the operator may choose to do something else,* there are 2N+1 alternatives among which to choose. We assume the operator's choice is a rational one governed by his expectations as to the system behavior, his goals and priorities, and his instructions. For example, performance priorities for the operator to achieve mission objectives in the RPV II study [2] were: i) pop-up (down) in good time sequence, ii) minimize ETA error at hand-off, iii) time phased RPV arrivals, iv) minimize lateral deviation errors, v) minimize command traffic (allow for possible "jamming"), and vi) minimize missed strikes.

The mission factors may be incorporated in an expected net gain criterion (see Section 2.4) of the form

$$\mathrm{ENGM}(i) = C_i P_i - CM_i \quad ; \; i=1,2,\ldots,2N$$

Here the constant C_i is the cost associated with the event that the state x_i exceeds some "tolerable" threshold TM_i, while P_i is the subjective probability of this event. As stated earlier, the information processor provides the subjective probability distribution for the states x_i of the RPV system under consideration, and, hence, the P_i.

* For example, monitor other RPV's not accounted for in the basic state space model in DEMON, such as those on the return list.

The constant CM_i is the action cost of monitoring the display y_i. Assuming that only a single displayed variable is selected for monitoring at a given time, the operator would monitor that y_i which has the greatest (positive) ENGM(i) associated with it. If none of the ENGM(i) is positive, no y_i will be monitored at that time which corresponds to the operator doing "other things".

The parameters in the expression for the expected net gain from monitoring are the threshold, TM_i, associated with the P_i, and the costs C_i and CM_i. These are the quantities that reflect mission objectives, etc. Assumptions about these parameters (and those associated with patching decisions) are analogous to some of the statements that are necessary for the Operator Attribute file in the Pritsker model (see Sec. 2.2).

There is considerable flexibility and some redundancy with respect to choices among these parameters. In particular, there is no requirement that they be constant in time. Thus, we can allow the monitoring thresholds to shrink linearly (or nonlinearly) with time to reflect the fact that a deviation of a given magnitude is much more important near pop-up than it is at launch. Alternatively, the threshold could be constant, reflecting an equal concern for errors throughout the mission. The relative magnitudes of C_i (or CM_i) may be chosen to emphasize the relative importance of ETA over LATDEV or of one RPV (say a strike) over another.

The CM_i can also be used to account for the importance or priority associated with pop-up or hand-off. This is accomplished by letting $CM_i = CM_{io} - CMP_i - CMH_i$. Here, CM_{io} is a constant, reflecting some basic monitoring cost that is fixed during the enroute phase. CMP_i is chosen to be zero from launch until τ_p seconds prior to the scheduled pop-up time T_p at which time it takes on a positive value. This value of CMP_i is kept constant until pop-up is completed. Similarly CMH_i has a value of zero until τ_H seconds prior to scheduled hand-off at which time it is given a positive value which is maintained until hand-off is complete. Note that the quantities τ_p and τ_H are closely related to corresponding parameters in the Operator Attribute file; they reflect operator preferences and/or instructions with respect to pop-up and hand-off.

3.2.3 Patching Strategy

By a patching strategy we mean the rational decision whether or not to issue a pop-up, hand-off or a patch (ETA or LATDEV) command. The patching strategy will depend on the monitoring strategy. For example, in the present implementation we assume that monitoring ETA is essential for making a velocity patch and that monitoring LATDEV is essential for making a LATDEV patch. It is assumed that pop-up or hand-off of an RPV may be done while monitoring either of its associated displays.

Just as for monitoring, we assume that the decision to patch is made on the basis of criterion functions that reflect mission goals, etc. Given that the operator is looking at the status display, the choice of a LATDEV patch, pop-up or hand-off command will depend on the expected net gain associated with these commands. If none of the expected gains are sufficiently large to offset patch inhibitions such as the desire to maintain radio silence, then no command will be issued. Otherwise, the command with the highest ENG will be executed. Similarly, for the patch decisions when looking at the menu display.

We formalize this notion again by means of relatively simple expressions that can be related to the task. Consider the case where the operator is looking at the status display. We define the expected net gain associated with each of his possible patch actions by the following:

$$\mathrm{ENGL}(i) = CL_i(\hat{x}_{2i})^2 - CLK_i$$

$$\mathrm{ENGP}(i) = CP_i \exp(t-T_P+\tau_P) \quad [\exp(2t_F)-1]$$

$$\mathrm{ENGH}(i) = CH_i \exp(t-T_H+\tau_H) \quad [\exp(2t_F)-1]$$

The rationale for selecting the LATDEV patching criterion ENGL(i) is as follows: $CL_i(x_{2i})^2$ is a measure of the cost of the lateral deviation x_{2i} of the i-th RPV from its flight plan. The purpose of a LATDEV patch is to reduce the lateral deviation x_{2i} to zero. It can be shown that, given the estimate $\hat{x}_{2i}$ and its uncertainty, the operator's expected gain from patching is simply $CL_i(\hat{x}_{2i}{}^2)$. As in the case for monitoring, this potential gain must be balanced against a patching cost and this is reflected in the CLK_i. The patching cost may be dependent on time and so CLK_i is not necessarily constant. A reasonable choice seems to be to have the inhibition against patching, hence CLK_i, decrease with time. Rewriting $ENGL(i) = CL_i[\hat{x}_{2i})^2 - TL_i^2]$, where $TL_i^2 = CLK_i/CL_i$, we may use the LATDEV objective of the mission to select the value TL_i at hand-off. The value of TL_i at launch is then a parameter to be selected.

Turning now to ENGP(i), the quantities T_p and τ_p were explained earlier while discussing the monitoring strategy. Note that the expression [exp $(2t_F)-1]$ is included to account for the two-frame $(2t_F)$ duration required for a pop-up. CP_i $\exp(t-T_p+\tau_p)$ is the assumed cost of missing a pop-up where the constant CP_i behaves similarly to CMP_i discussed earlier. That is, it has a value zero until τ_p seconds prior to pop-up at which time it takes on a positive constant value until pop-up is completed. Although this positive constant may be used as a parameter of the model, we may also choose CP_i by rationalizing the cost of missing a pop-up against that of losing an RPV because of excessive LATDEV, i.e. by setting the costs equal and solving for CP_i. Of course, the same arguments apply to the ENGH(i).

The identical problem faces the operator when viewing the menu display; only here the ENG from a velocity patch must be compared with ENGP(i) and ENGH(i). The expected net gain for a velocity patch is defined analogously to that for a LATDEV patch, namely by

$$\mathrm{ENGV}(i) = \mathrm{CV}_i \ (\hat{x}_{2i-1}/V_i)^2 - \mathrm{CVK}_i = \mathrm{CV}_i [(\hat{x}_{2i-1}/V_i)^2 - \mathrm{TL}_i^2$$

with the choice of CV_i and CVK_i governed by similar considerations.

In summary, while monitoring LATDEV, based on his expectations the human operator (in the DEMON implementation) will decide to pop-up, hand-off or do a LATDEV patch depending on which of the expected net gains is most positive. If none of them is positive he will not patch, pop-up, or hand-off. Note that his decision **not** to patch may depend on his concern for breaking radio silence (as reflected in the shrinking "threshold" TL_i). Similarly, while monitoring ETA he may decide to pop-up, hand-off, do a velocity patch or preserve radio silence.

3.2.4 Patch Command Generator

Once a decision is made to patch a particular RPV, it is necessary to compute and execute the patch control. The purpose of a patch control is to guide the RPV from its current location and heading to intercept and fly along the planned flight path. We assume a simple strategy of minimizing the time to return to the planned flight path assuming that the two control actions on a given RPV are non-interactive. In this case the control actions are trivially computable using the current estimate of the ETA and LATDEV states:

$$u_i = -\hat{x}_i/T$$

where T is the duration over which the patch is to take place. To relate the u_i to the velocity and heading we recall from Section 3.1 that

$$u_{2i-1} = V_i \cos \phi_i - \bar{V}_i$$
$$u_{2i} = V_i \sin \phi_i$$

which shows the interaction of the velocity patch V_i and the LATDEV patch ϕ_i. Note that in the two-state formulation of the RPV problem there is no true heading. However, we use the above equations to define "pseudo constant headings" ϕ_i during the LATDEV patch.

The velocity patch on RPV-i will be computed as

$$V_i = (u_{2i-1} + \bar{V}_i)/\cos \phi_i$$

where

$$u_{2i-1} = -\hat{x}_{2i-1}/T$$

A check is made on V_i to see of it is within the allowable limits. If it exceeds the limit then V_i is reset at the limit by adjusting the patch time T and hence the patch control action u_{2i-1}. Then u_{2i} is adjusted to account for its interaction with the velocity patch.

A LATDEV patch will be computed as a constant pseudo heading

$$\phi_i = \text{ARCSIN}\ (u_{2i}/V_i)$$

where

$$u_{2i} = -\hat{x}_{2i}/T$$

A "pseudo radius of turn" R is computed as described earlier in Section 3.2. If R violates the specified minimum turn radius RMIN for the RPV then the patch time T is relaxed to satisfy the turn radius constraint. To reflect the operator's experience on making LATDEV patches, a factor called SFACTR is introduced. SFACTR ranges from .2 to 1 and is nominally 1. It is decremented in steps of .2 for each consecutive disallowed LATDEV patch and incremeted in steps of .1 for each consecutive successful LATDEV patch. The operator will use SFACTR as a safety factor to avoid tight turns for the LATDEV patch and will use RMIN/SFACTR as a guide to select the radius of turn for the LATDEV patches. Having decided on a patch time T, the control action u_{2i} is recomputed and then u_{2i-1} adjusted to reflect the effect of LATDEV on u_{2i-1}.

3.3 Implementation of the Model

DEMON, the decision making, monitoring, and control model of the human operator is implemented in FORTRAN. The program has a modular structure to facilitate ease of adding further modules to include alternative monitoring, control, and decision strategies that may appear promising at a future date.

To accommodate the random aspects of the problem, the program basically has a Monte-Carlo simulation character. The specialized version of DEMON for the RPV problem will produce as outputs the "true" time-histories of the RPV flights, the sequence of monitoring and patching decisions made, and the resulting performance (samples of outputs are included in Section 4).

The important aspects of the simulation program implementing Demon are shown in the flow diagram in Figure 5. There are, as indicated, nine major modules in the program. Modules 4, 6 and 7 are of special interest because they do not arise in the usual manual control models. The theory behind these modules is developed in Section 3.3 (also see reference [1]; Appendices A and B).

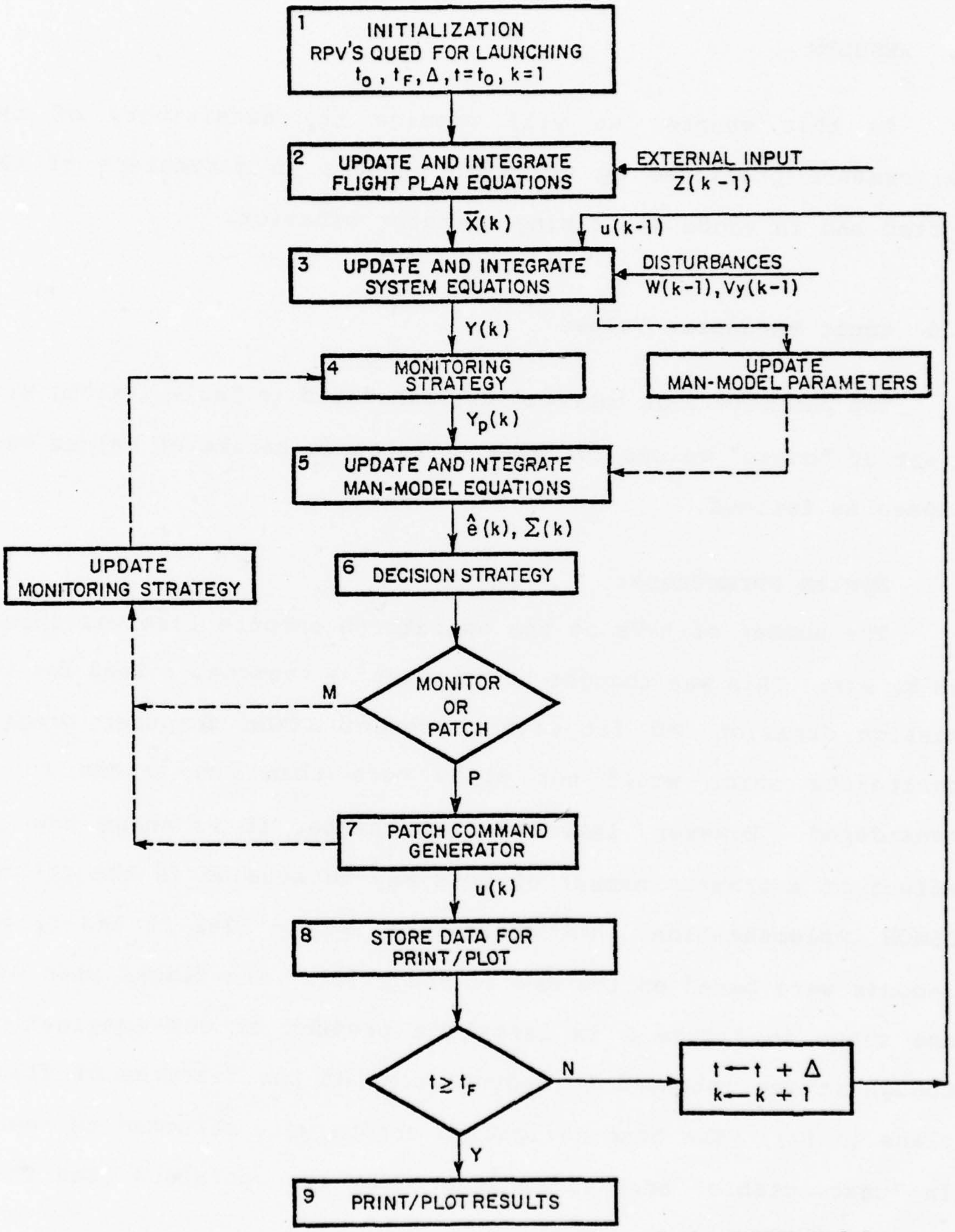

1 INITIALIZATION RPV'S QUED FOR LAUNCHING
t_0, t_F, Δ, $t=t_0$, k=1
2 UPDATE AND INTEGRATE FLIGHT PLAN EQUATIONS
EXTERNAL INPUT Z(k-1)
$\overline{X}(k)$
u(k-1)
3 UPDATE AND INTEGRATE SYSTEM EQUATIONS
DISTURBANCES W(k-1), Vy(k-1)
Y(k)
4 MONITORING STRATEGY
UPDATE MAN-MODEL PARAMETERS
$Y_p(k)$
5 UPDATE AND INTEGRATE MAN-MODEL EQUATIONS
$\hat{e}(k)$, $\Sigma(k)$
UPDATE MONITORING STRATEGY
6 DECISION STRATEGY
MONITOR OR PATCH
M
P
7 PATCH COMMAND GENERATOR
u(k)
8 STORE DATA FOR PRINT/PLOT
$t \geq t_F$
N
t ← t + Δ
k ← k + 1
Y
9 PRINT/PLOT RESULTS

Fig. 5 Flow Chart for Computer Implementation of the DEMON Model

4. RESULTS

In this chapter we will examine the sensitivity of the performance predicted by DEMON to changes in parameters of the system and in those describing operator behavior.

4.1 Basic Parameter Values

The parameters of interest are exhibited in Table 3 along with a set of "basic" values for them. The basic parameter values were chosen as follows.

System Parameters:

The number of RPVs on the operator's enroute list was chosen to be six. This was thought to represent a reasonable load for the mission duration and fit within present DEMON computer program contraints which would not allow more than two triads to be considered. However, later in this chapter it is shown how the effect of a greater number of RPVs may be studied in the present DEMON implementation. The choices of R_{min} = 528Ø ft and t_f = 5 seconds were based on the RPV II study [2]. The flight plan used and shown in Figure 2 is largely a product of our imagination, though it was intended to capture some of the features of flight plans in [2]. The base navigation errors were selected to result in "unacceptable" errors, if ignored by the operators (see Table

ID	SYMBOL	DESCRIPTION	VALUES FOR BASIC CASE
	SYSTEM	CHARACTERISTICS	
S1	N	Number of RPV's (RPV "density")	6
S2	R_{MIN}	Minimum allowable turn radius	5280 ft.
S3	t_F	Display update rate	5 sec.
S4	FLTPLN	Flight Plan Parameters such as flight duration and desired time for launch, turns, pop-up, hand-off, turn angles, etc.	Figure 2
S5	w_i	Navigation errors	Eq. (4)
S6	v_{y_i}	Reporting error	-20 dB
HUMAN OPERATOR CHARACTERISTICS			
H1	TM_i	Monitoring threshold	250 ft. LATDEV 5 sec. ETA
H2	C_i	Cost of exceeding TM_i	1
H3	CM_i	Monitoring Action Cost	0
H4	CL_i	Cost factor for LATDEV	1
H5	TL_i	LATDEV Patch cost	250
H6	CV_i	Cost factor for ETA deviation	109375
H7	TV_i	Velocity patch cost	3375
H8	CP_i	Pop-up cost factor	$2.19E7/Exp(\tau_p)$
H9	τ_p	Operator preferred pop-up interval	5 sec.
H10	CH_i	Hand-off cost factor	CP_i
H11	τ_H	Operator preferred hand-off interval	5 sec.
H12	WØM	Operator's understanding of system navigation errors	101250 10000

Table 3 Important Parameters in the DEMON Model

2). The value of -20dB for the noise-to-signal ratio associated with reporting errors was chosen arbitrarily, but corresponds to good "reporting" performance.

Human Operator Parameters:

The parameters of the operator are essentially those associated with the expressions for expected net gains. These were picked on the basis of the following mission related considerations. The choice of 3375 ft. and 250 ft. for the monitoring thresholds was based on the assumption that the operators are instructed that a LATDEV of 250 ft. and an ETADEV of 5 sec. are tolerable. The choice of C_i=1 and CL_i=1 are made because these may be used as normalizing factors without affecting the operator's decision strategies. The monitoring action cost of CM_i=0 corresponds to an operator who is completely dedicated to the N=6 RPVs in his enroute list. The patch costs TL_i=250 and TV_i= (5 times 675) reflect the assumed mission objectives of 250 ft. LATDEV and 5 sec. ETADEV at handoff. The choice of TP_i relative to CL_i=1 is made by rationalizing a LATDEV patch with a pop-up. For example, the operator is assumed to consider a LATDEV patch to be as important as a pop-up if the LATDEV near pop-up time is 1500 ft. The rationale here is that if LATDEV exceeds 1500 ft. then terminal control of that RPV will be lost. Equating the ENGL and ENGP in this case results in the choice of CP_i shown. Similar argument yields the values for CV_i and CH_i. The values $\tau_P=5$ and $\tau_H=5$ correspond to a frame time .

The covariance of the noise (WØM) associated with the internal model of equation (8) is a different type of oprator parameter. As noted previously, they are intended to reflect the operator's expectations concerning the navigation errors as determined from training or instruction. The values shown were selected to yield a probability of .9 of exceeding ETA and LATDEV thresholds of 5 sec. and 250 ft., respectively, at the time these thresholds were actually exceeded with the nominal navigation errors.

Several DEMON runs were made to ascertain the sensitivity to various model parameters. These sensitivity results are described below.

Figure 6 shows an example of the 'activity chart' output during an exercise of the DEMON model. This activity chart shows all monitoring and patching activity that goes on during the mission. The chart shows that the activities are governed by the instantaneous values of the states and their estimates, and are not preordained before the model run to occur in any particular sequence. The first column under RPV 1 shows the monitoring decision M, S or a blank according to whether the Menu, status or no display was monitored. The next column shows the patching

```
STARTING DEMON - VERSION 17
INPUT READ FROM FILE: DM6R1 .INP

DEMON SIMULATION OF A 2-STATE 6 RPV PROBLEM
     CM = .25 , N = 6 , (NOISE WØM=648.E4,6.4E4)
 TIME  RPV 1  RPV 2  RPV 3  RPV 4  RPV 5  RPV 6
  Ø.  M
  5.  S
 1Ø.  M
 15.  S
 2Ø.  M
 25.  S
 3Ø.  M
 35.  S
 4Ø.  M
 45.         S
 5Ø.         M
 55.  S
 6Ø.         S
 65.  M
 7Ø.  S
 75.         S
 8Ø.         M
 85.  SL
 9Ø.     GO          S
 95.                 M
1ØØ.         S
1Ø5.  M
11Ø.  S
115.                S
12Ø.         M
125.         SL
13Ø.  S         GO
135.                       S
14Ø.                       M
145.                S
15Ø.                M
155.         S
16Ø.  SL
165.  M  GO
17Ø.                       S
175.         MV
18Ø.                              S
185.                              M
19Ø.                SL
195.         SL        GO
2ØØ.  SL        GO
2Ø5.     GO                M
21Ø.                       SL
215.                M         GO
22Ø.                              S
225.                                     S
23Ø.                                     M
```

Fig. 6 Example Activity Chart for a Simple RPV Mission

```
490.                       SL
495.                          GO         MV
500.  S
505.                              SL
510.  MV                             GO
515.                                     SL
520.         M                              GO
525.                SL
530.         S         GO
535.                              MV
540.                       S
545.  SL
550.     GO                MV
555.                              S
560.                                     S
565.                M
570.                S
575.         SL
580.            GO                       MV
585.                       SL
590.  S                       GO
595.  M
600.                              SL
605.         MV                      GO
610.                                     SL
615.                SL                      GO
620.                   GO         M
625.         S
630.                       S
635.                       MV
640.  SL
645.     GO         MV
650.                              S
655.                                     S
660.                S
665.                                     MV
670.         SL
675.            GO         SL
680.  MV                      GO
685.  S
690.         M
695.                              SL
700.                              MV GO
705.                                     SL
710.                SL                      GO
715.         S         GO
720.                       M
725.                       S
730.                M
735.  SL
```

Fig. 6   continued

```
235.                S
240.         S
245.  S
250.  M
255.                       S
260.         MV
265.                              SL
270.                              M  GO
275.                                     S
280.                SL
285.         SL        GO
290.            GO         MV
295.  SL
300.     GO         M
305.                       SL
310.                          GO  S
315.                                     M
320.                                     SL
325.                S                       GO
330.         S
335.                       SL
340.                       SL GO
345.  MV                      GO
350.  SL
355.     GO  M
360.                              SL
365.                              M  GO
370.                                     S
375.                S
380.                       MV
385.         SL
390.            GO         S
395.                M
400.  S
405.                                     MV
410.                              S
415.                                     SL
420.                SL                      GO
425.  MV               GO
430.         S
435.         MV
440.                       S
445.  SL
450.     GO                       M
455.                              S
460.                                     SL
465.                                     S  GO
470.                       M
475.                S
480.         SL
485.            GO  MV
```

Fig. 6 continued

```
740.     GO                         S
745.                                       M
750.                 SL
755.                 S  GO
760.                                       S
765.          SL
770.  M          GO
775.                        SL
780.          MV               GO
785.  SL
790.     GO                         SL
795.                                M  GO
800.                 S
805.                        MV
810.                                       SL
815.          S                               GO
820.                 M
825.                        S
830.                                       MV
835.  S
840.                                S
845.                 SL
850.  M                 GO
855.                                       S
860.          SL
865.          M  GO
870.                        SL
875.                           GO   MV
880.  SL
885.     GO                         SL
890.                                SL GO
895.                                S  GO
900.                 S
905.                        M
910.                                       SL
915.          S                               GO
920.                 M
925.                        SL
930.                           GO          M
935.  SL
940.     GO   MP
945.  M
950.                                SL
955.  SP                               GO
960.                 SL
965.          SH        GO
970.                                       SL
975.                                M         GO
980.  SH
985.                        SL
```

Fig. 6 continued

```
 990.                       MV GO
 995.                              S
1000.                M
1005.                S
1010.                                     M
1015.                                     S
1020.                       S
1025.                              SL
1030.                SP               GO
1035.                              M
1040.                       M
1045.                                     SL
1050.                       SL               GO
1055.                MH        GO
1060.                              S
1065.                                     M
1070.                                     S
1075.                              MP
1080.                       S
1085.                              SL
1090.                       MP        GO
1095.                                     S
1100.                              SH
1105.                       SL
1110.                          GO         M
1115.                       MH
1120.                                     SL
1125.                                     M  GO
1130.                                     S
1135.                                     M
1140.                                     S
1145.                                     M
1150.                                     S
1155.                                     M
1160.                                     S
1165.                                     M
1170.                                     SP
1175.                                     M
1180.                                     S
1185.                                     S
1190.                                     MH
1195.
1200.
1205.
1210.
1215.
```

Fig. 6 continued

decision H, P, V or L. If a LATDEV patch decision was made, the result of the GO/NOGO check made by the system in the next frame is also indicated. Similarly the monitoring and patching activity for all the RPVs are indicated. It can be seen that early in the run, the activities concentrate on the RPVs already launched and build up in intensity as the mission progresses. The activities taper off towards the end of the enroute mission as the RPVs are handed-off. Subsequent results are intended to show overall trends of the effects of parameter changes. These are obtained from a set of summary statistics output by the implementated DEMON model.

4.2 Monitoring Performance

The DEMON model was exercised on several selected values for the parameters characterizing the human operator. Figure 7 shows the effect of varying the monitoring cost CM_i on the monitoring frequency. Two curves are shown. For the curve labelled TM = 1:1, the monitoring thresholds TM_i are constant from launch to hand-off. For the curve labelled TM = 4:1 the monitoring thresholds at launch are four times what they are at hand-off and decrease linearly with mission time. It is seen from Figure 7 that the effect of increasing monitoring cost is to decrease the monitoring frequency (averaged over all the 6 RPVs under consideration). For any given monitoring cost CM, widening the monitoring threshold also decreases the monitoring frequency.

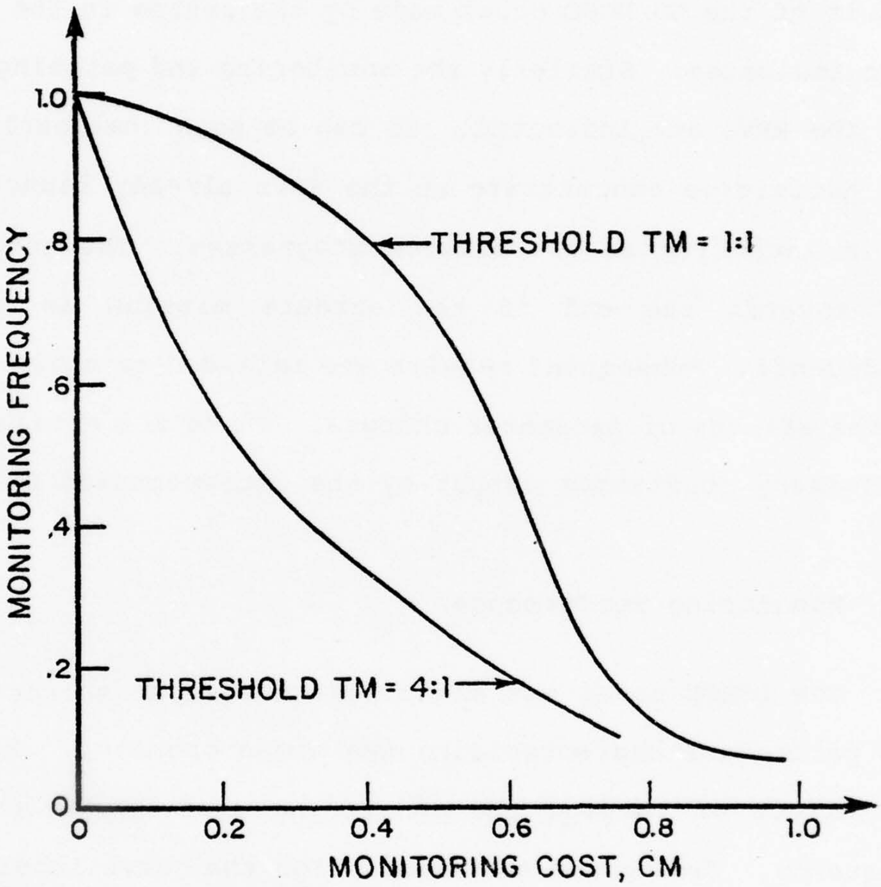


Fig. 7 Effect of Monitoring Cost CM (and Threshold TM) on Monitoring Frequency

Figure 8 shows the effect on the TM = 1:1 curve in Figure 6 if the operator assumes a different model WØM for the navigation errors. Recall that the ENGM expression (with C_i=1) depends on the difference P_i-CM_i. The subjective probabilities P_i depend both on the estimates $\hat{x}_i$ and the uncertainties σ_i associated with these estimates. A lower value of WØM couses lower rate of growth of σ_i and hence it takes longer for the P_i associated with different RPVs to be 'equalized' and start overwhelming the monitoring cost CM_i. Thus for the low WØM case, the effect of CM_i is more pronounced than in the high WØM case. In fact, it can be seen from Figure 8 that for any fixed monitoring cost CM, the monitoring frequency is higher if the operator expects the navigation errors to be greater (as represented by the higher noise covariance WØM). The lower value of WØM also reduces the effect of the pop-up cost, CMP_i. This results in missing scheduled pop-up times as indicated in Table 4 (to be introduced later).

The results in Figures 7 and 8 were obtained by using the same value of CM_i for each of the 6 RPVs. It was stated earlier in Section 3.2 that CM_i may be used to distinguish between RPVs. This is illustrated in Figure 8 which shows the combind effect of shrinking threshold and different RPV priority (in terms of CM_i). These results were otained in a further simplified, single state, formulation of the RPV problem with N=3 (see [4] for a detailed

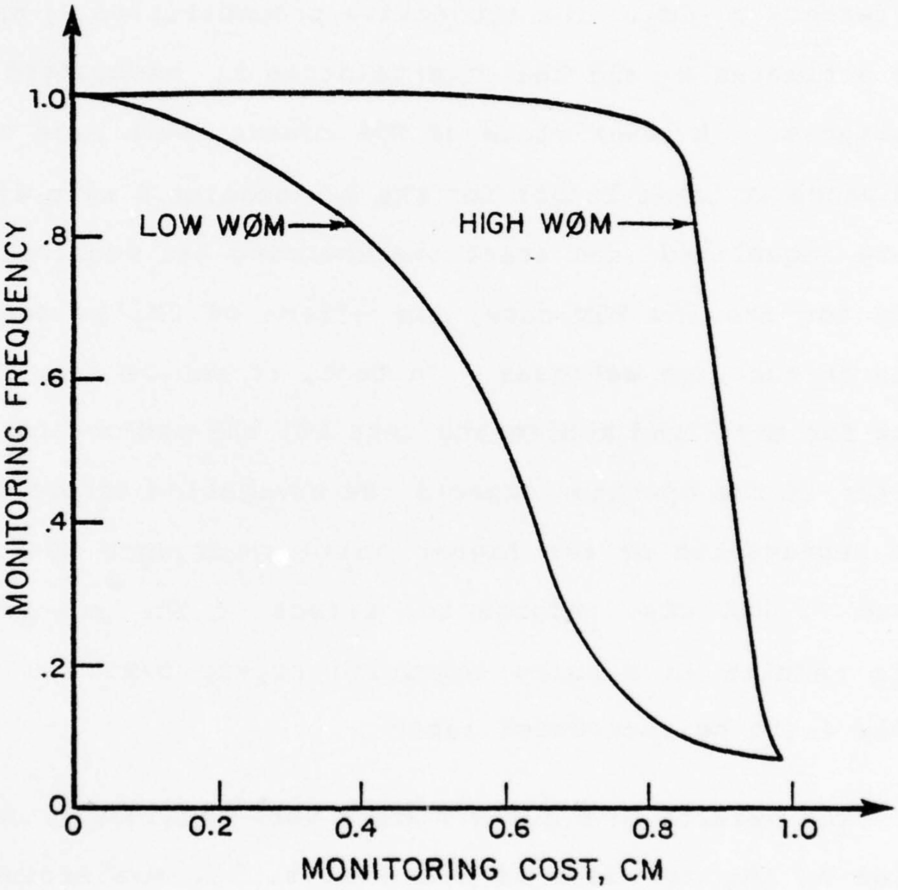


Fig. 8 Effect of Pseudo Process Noise WØM on Monitoring Frequency

Table 4 Error in Pop-Up Time

RPV #	1	2	3	4	5	6	Average
$\tau_p = 5$	55	10	5	0	5	0	12.5
$\tau_p = 2$	55	10	5	0	5	0	12.5
$\tau_p = 0$	50	10	5	5	40	5	19.1

description of this problem). The histogram plot of monitoring frequency in Figure 9 indicates that as mission time increases RPV monitoring frequency increases. But there comes a time when monitoring resouces are not adequate to satisfy the increasing needs of each of the RPVs and then the highest priority RPV demands most of the attention it can get while the lowest priority RPV gets no attention from the operator.

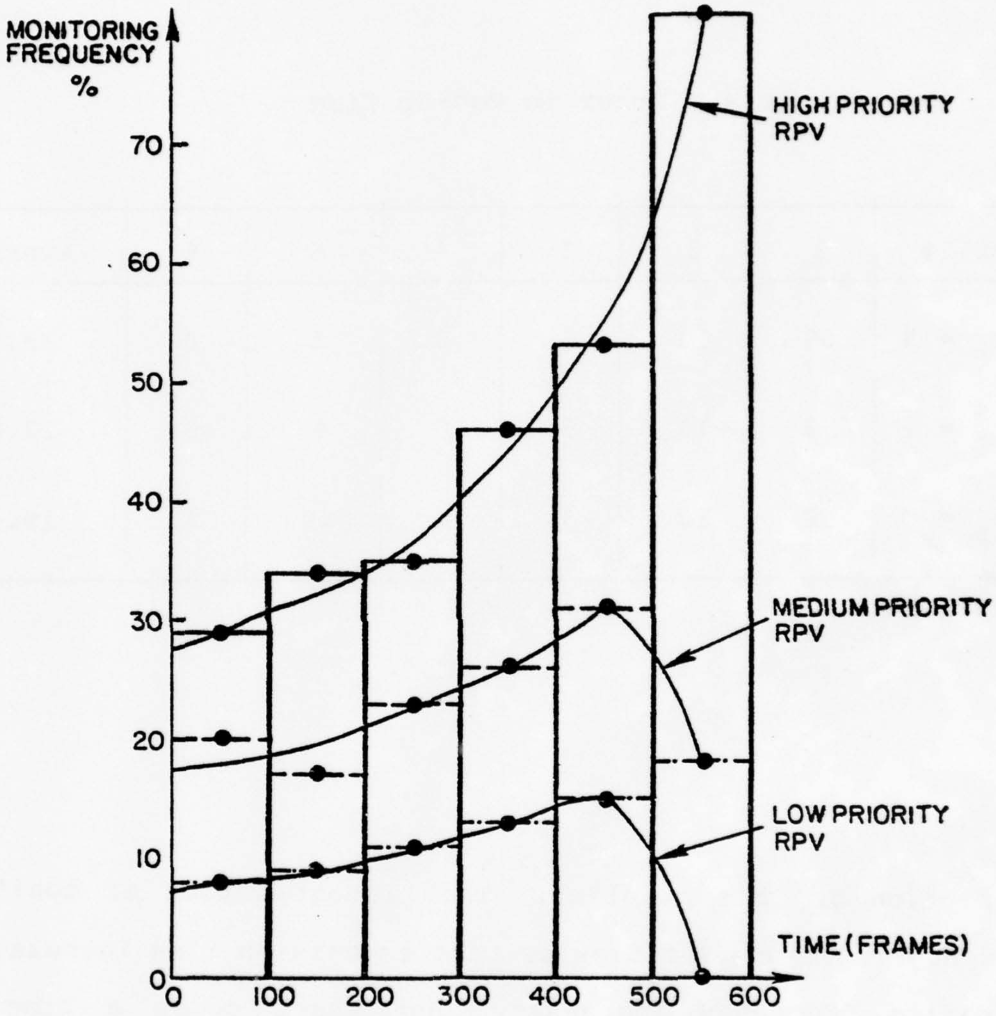


Fig. 9 Histogram Plot Showing Combined Effect of Shrinking Threshold and Different Monitoring Costs on Monitoring Activity

The effect of adding more RPVs on the operator's monitoring behavior is illustrated in Figure 10. In this simple excercise of the DEMON model, the RPVs in the operator's "enroute list" were increased from 1 to 6 while the pseudo random navigation error sequence driving RPV 1 remained unaltered. The monitoring looks on the S and M displays of RPV 1 are plotted against "workload" of the operator as measured by the number of RPVs in his list. Clearly, when the other RPVs compete for attention, the monitoring activity on RPV 1 decreases.

The present implementation of the DEMON model can only accomodate a maximum of 7 RPVs. To investigate the effect of a higher number of RPVs we can introduce the concept of an "equivalent RPV". We saw in Figure 10, that additional RPVs compete for monitoring attention. Likewise in Figures 7 and 8, the effect of increasing the monitoring cost CM is to decrease the monitoring frequency. Recall that CM can be used to account for the necessity of doing other things. High enough values of CM will result in periods in which the six basic RPVs are unmonitored. If we assume that these periods are devoted to other RPVs, and that all RPVs share monitoring attention approximately equally, then an "equivalent" number of RPVs is obtained by dividing the actual number under control by the fraction of the total time that they are being observed. For example, if a value of CM is chosen such

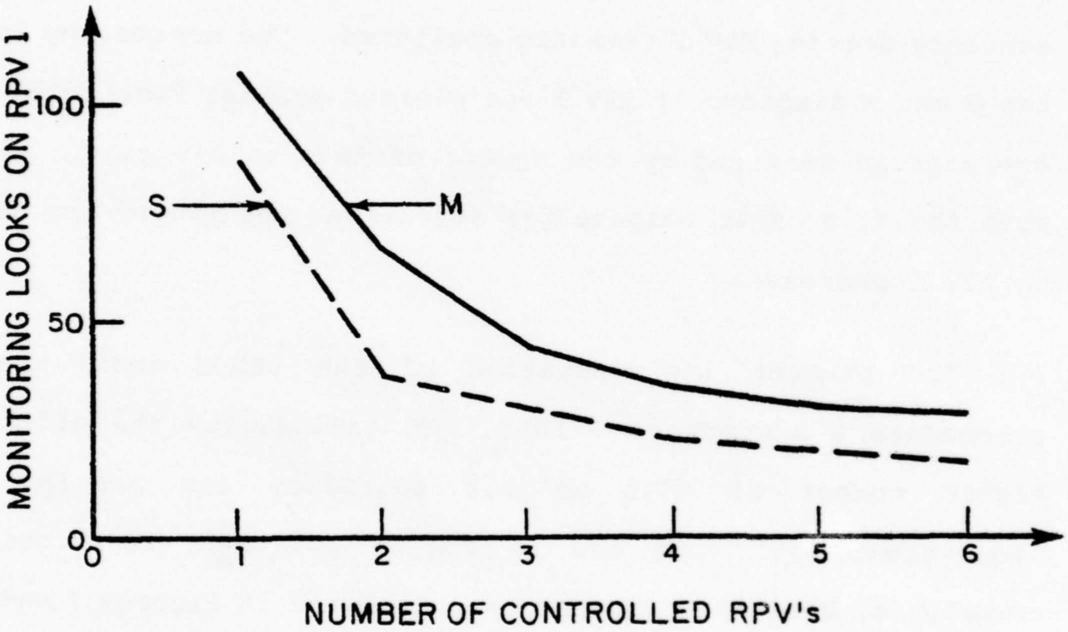


Fig. 10 Effect of Workload (Number of Controlled RPVs) on Monitoring Looks on RPV 1

that the basic six RPVs are monitored only 60 per cent of the time, then we can interpret this as equivalent to ten RPVs under control.

Using the concept of "equivalent RPVs" Figures 7 and 10 are combined to obtain the effect of workload (equivalent over the number of real RPVs in the operator's list). The result plotted in Figure 11 shows that average monitoring frequency is roughly proportional to the reciprocal of the equivalent number of RPVs.

4.3 Patching Performance

In Section 4.1, we considered the sensitivity of monitoring performance to the operator parameters CM, TM, WØM and the system parameter N. We now turn to the study of the operator's effectiveness in controlling the system performance.

The effect of varying patch costs is illustrated in the computer generated graphs of ETA deviation for RPV 1. Figure 12(a) is for the case of flat TL_i, TV_i and Figure 12(b) is for the case of TL_i shrinking to a value at hand off the same as that for the flat TL_i, starting from a value at launch equal to four times that at hand-off. Clearly, higher patch costs during the early part of the mission inhibit patches and result in greater ETA deviations. However, the shrinking TL_i ensures sufficient patching activity to keep ETA near hand-off to be comparable to that in the flat TL_i case.

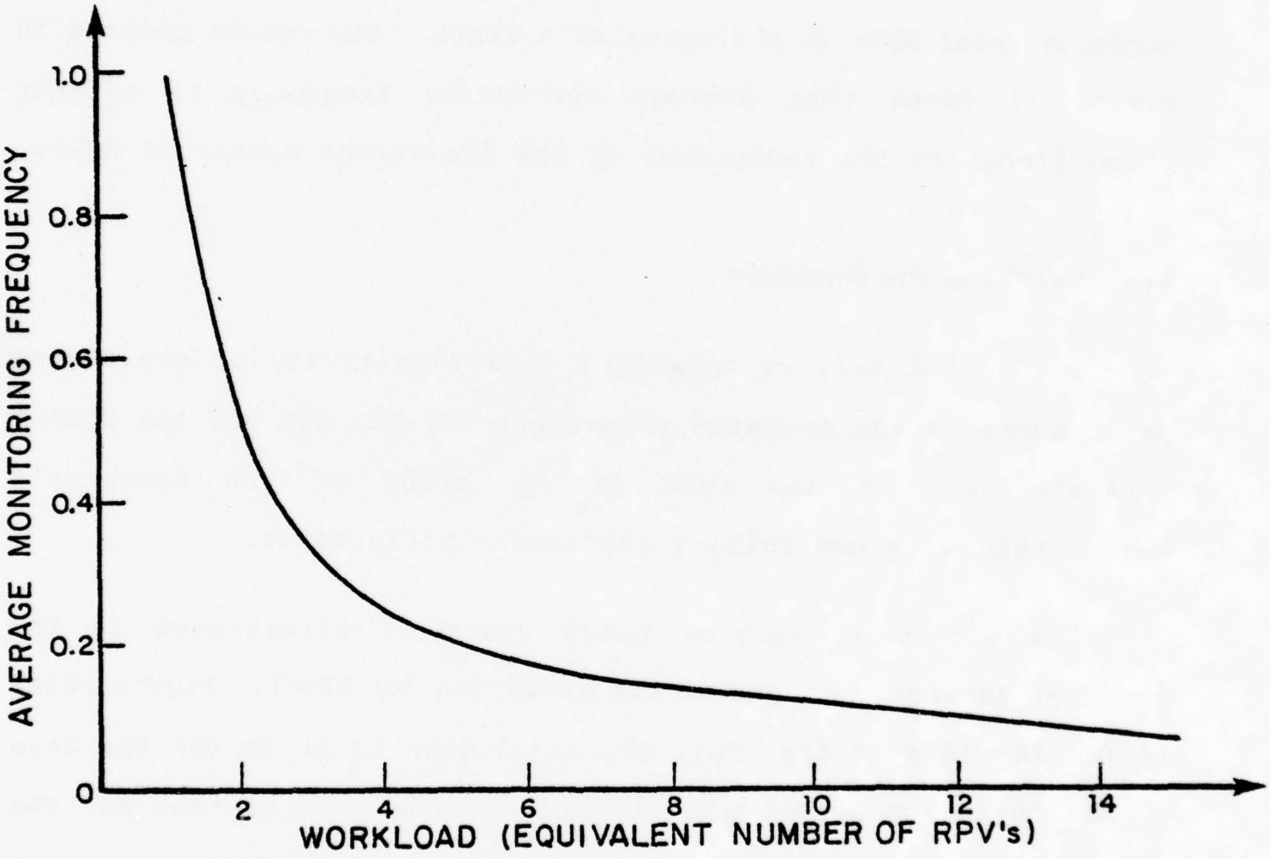


Fig. 11 Effect of Workload (Equivalent Number of RPVs) on Average Monitoring Frequency

]

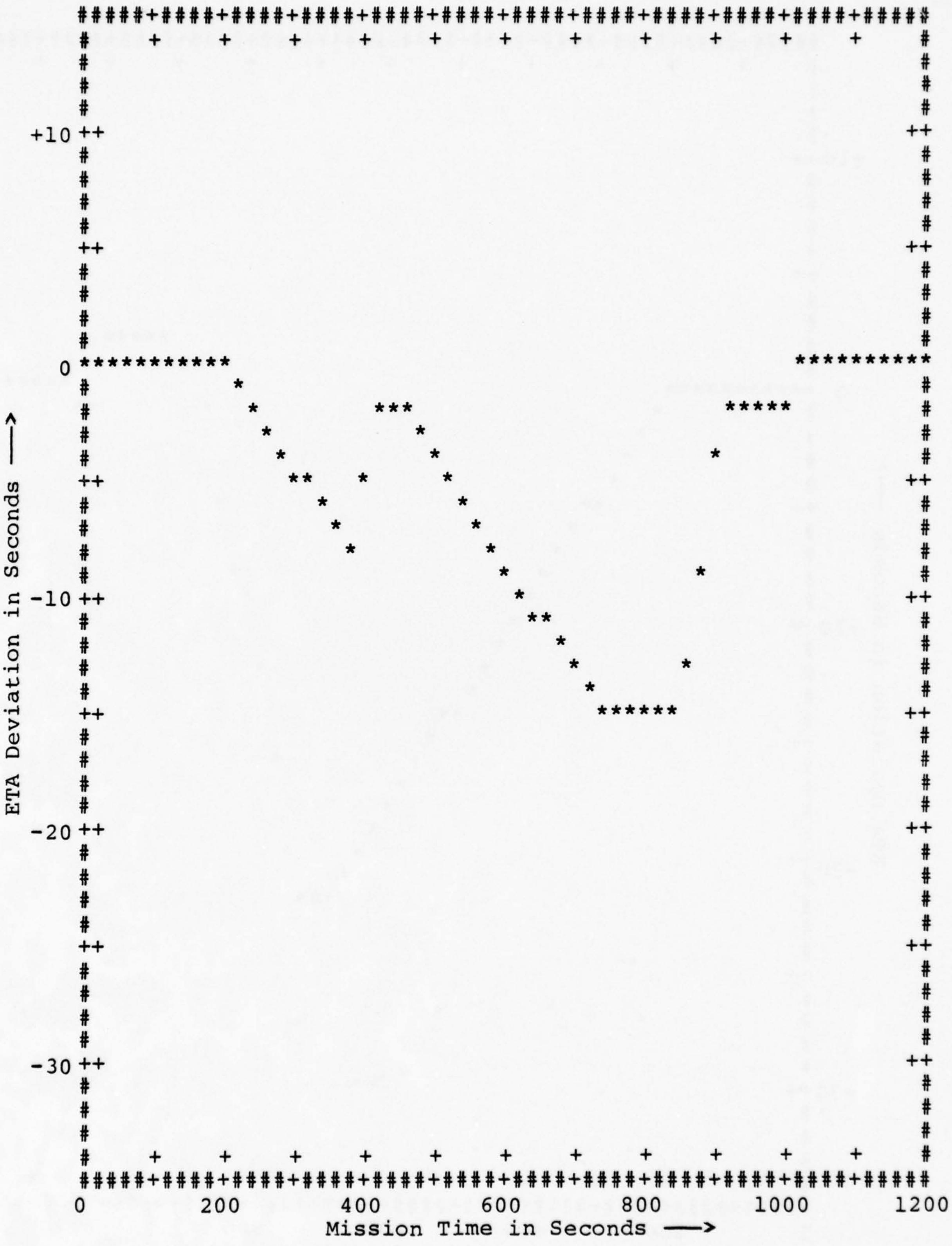


Fig. 12 a) The Effect of Varying Patch Costs on the ETA State Trajectory of RPV 1

i) Cost at Launch = Cost at Hand-off

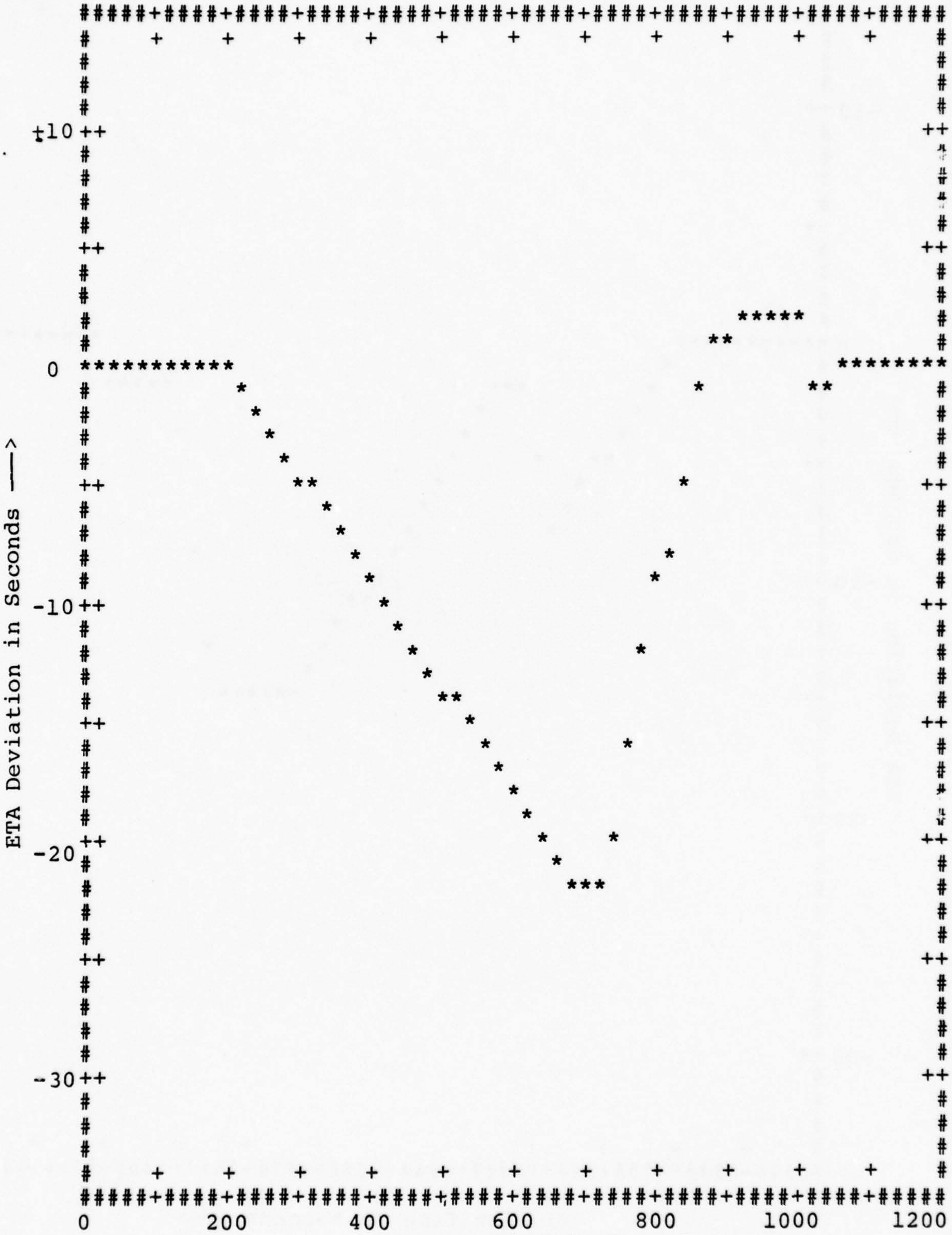


Fig. 12 a) The Effect of Varying Patch Costs on the ETA State Trajectory of RPV 1
ii) Cost at Launch = 2 Times Cost at Hand-off

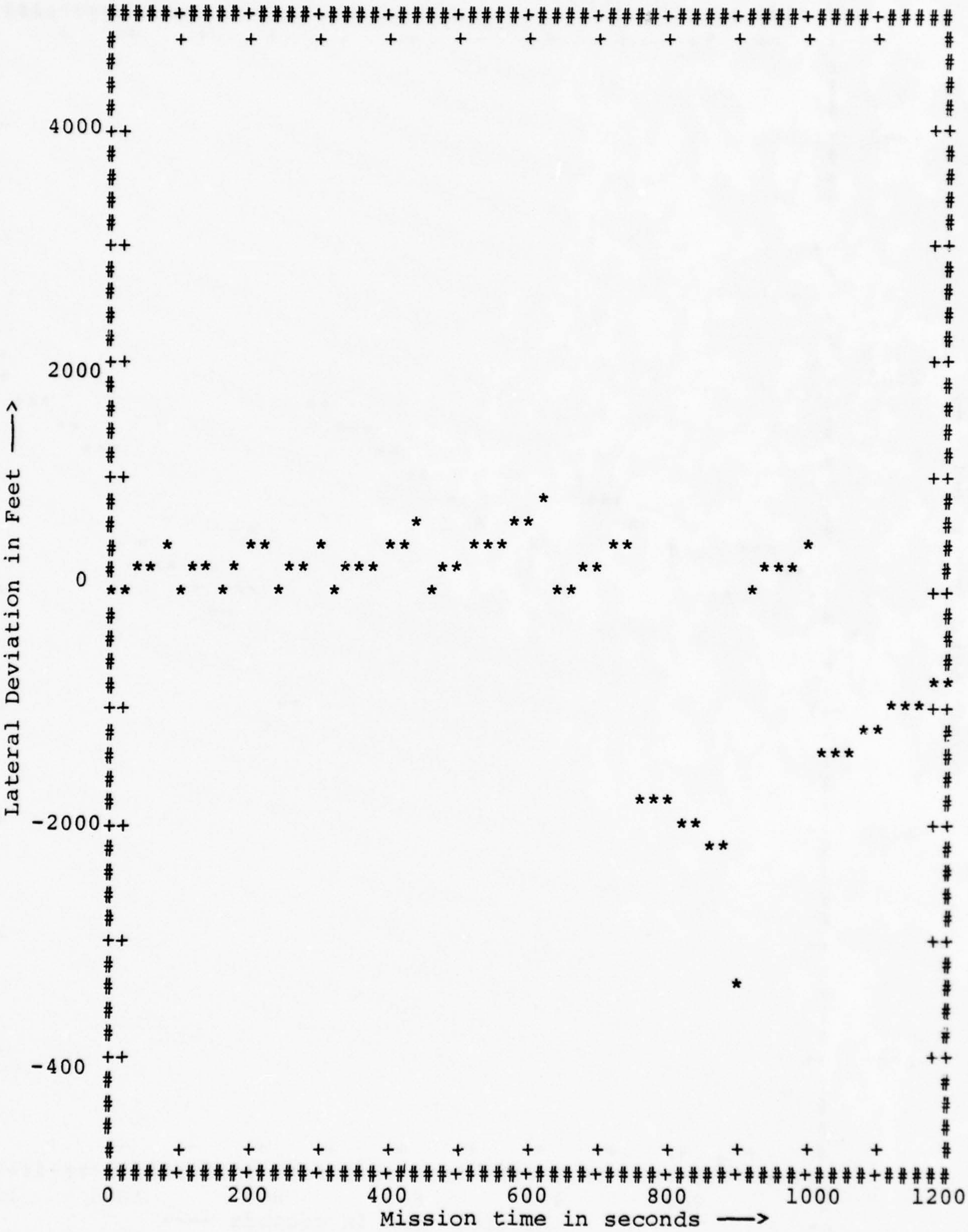


Fig. 12 b) The Effect of Varying Patch Costs on the LATDEV State Trajectory of RPV 1

i) Cost at Launch = Cost at Hand-off

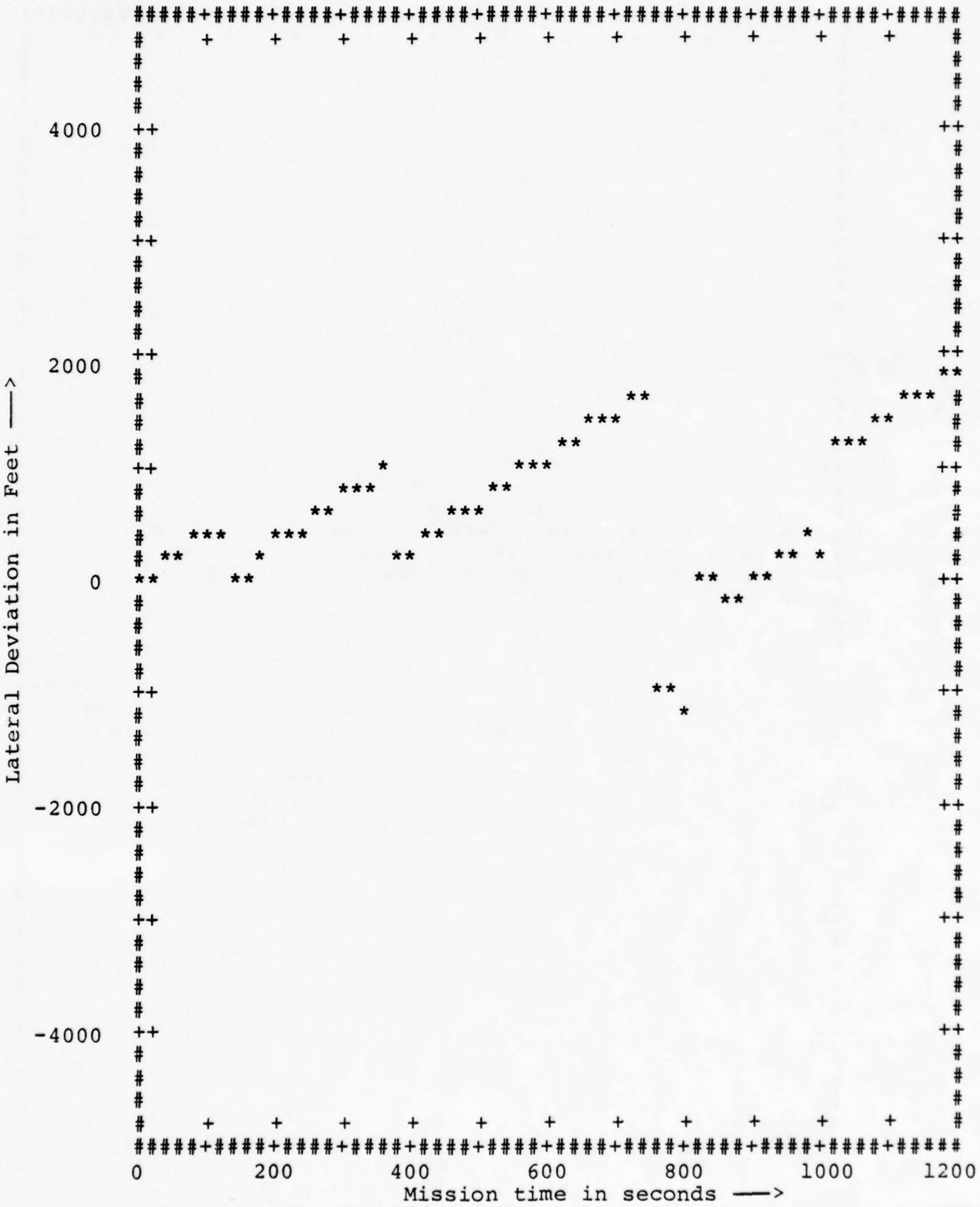


Fig. 12 b) The Effect of Varying Patch Costs on the LATDEV State Trajectories of RPV 1
ii) Cost at Launch = 2 times cost at Hand-off

Figure 13 shows the effect of workload (in terms of equivalent number of RPVs) on the normalized controlled RPV performance as defined below. RPV performance is measured in terms of the RMS values of the deviations (ETADEV and LATDEV) from the flight plan. The RMS values are obtained by "time averaging" and thus include errors all along the flight plan from launch up to hand-off. A controlld RPV is a real RPV on the operator's enroute list. The RMS value was averaged over the controlled RPVs and then divided by the indifference threshold setting (chosen to be 25Ø' for LATDEV and 5 sec. for ETADEV in the model run) to obtain the normalized controlled RPV performance. The results for LATDEV and ETADEV are plotted on Figure 13. *The operator is able to control* up to four RPVs without a degradation in performance. Then, performance steadily deteriorates as workload increases. It is clear that for the chosen parameter settings there is a critical point at N=7. The average RMS LATDEV, integrated from launch to hand-off for N=7, has a value of roughly 15ØØ' with performance deteriorating at a high rate.

It is interesting to note that the RMS errors seem to flatten out for N>15. A further analysis shows that these peak values of RMS errors are the errors inherent in the mission due to navigation errors, that is, the errors that would result if the operator's enroute list were "empty" and the RPVs try to negotiate the flight plan without control.

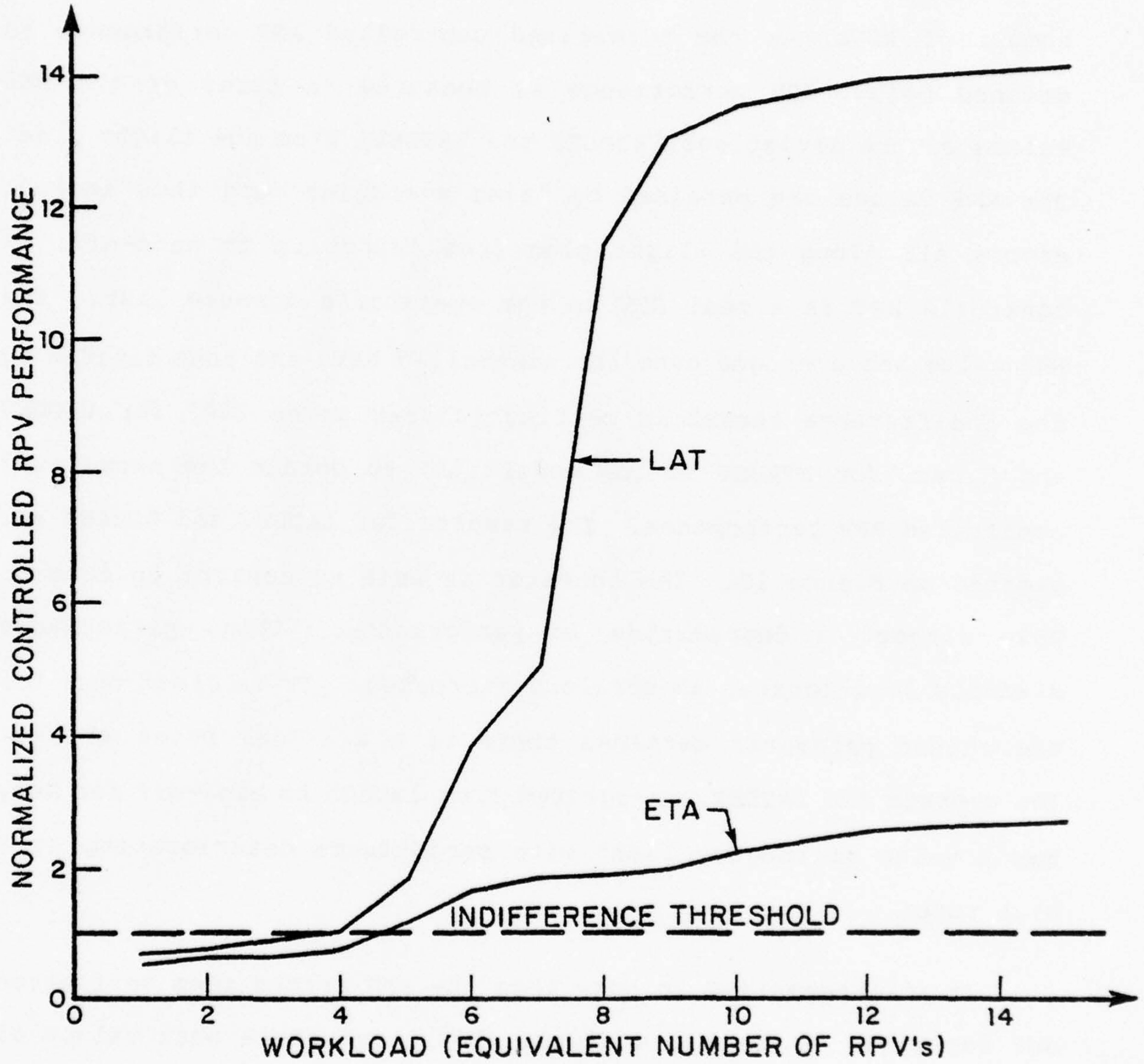


Fig. 13 Effect of Workload (Equivalent Number of Controlled RPVs) on Normalized Controlled Performance for Low WØM.

The results shown in Figure 13 were obtained with the operator's view of the navigation error as a "low" noise process. The same model runs were repeated, this time with the operator's view of the navigation errors to be a "high" noise process. The resulting curves (Figure 14) show that the RMS errors are lower than in the "low" noise case. This is because the higher noise causes the monitoring frequency to increase (see Figure 8) thus improving the operator's estimation process.

Now we turn to the effect of τ_p on pop-up. Recall that τ_p seconds prior to the desired pop-up time T_p, the DEMON operator begins to be concerned about pop-up. A few model runs were made by varying τ_p and the results are displayed in Table 4. As it was mentioned earlier the lower WØM causes the DEMON operator to miss pop-ups even with a τ_p of 5 sec. It is seen from Table 4 that with $_p$=Ø the average error in pop-up time is even larger. These errors in pop-up time also cause similar errors in hand-off time which increases from an average value of 12.5 for τ_p=5 to 31.7 for τ_p=Ø. Note that the 5 sec. frame update time makes all intermediate values of τ_p between Ø and 5 behave the same as $_p$ = 5 sec.

The effects of various human parameters on monitoring and patching performance were discussed above. The only system parameter that was included in the above discussion was the workload as represented by the number N of RPVs for which the

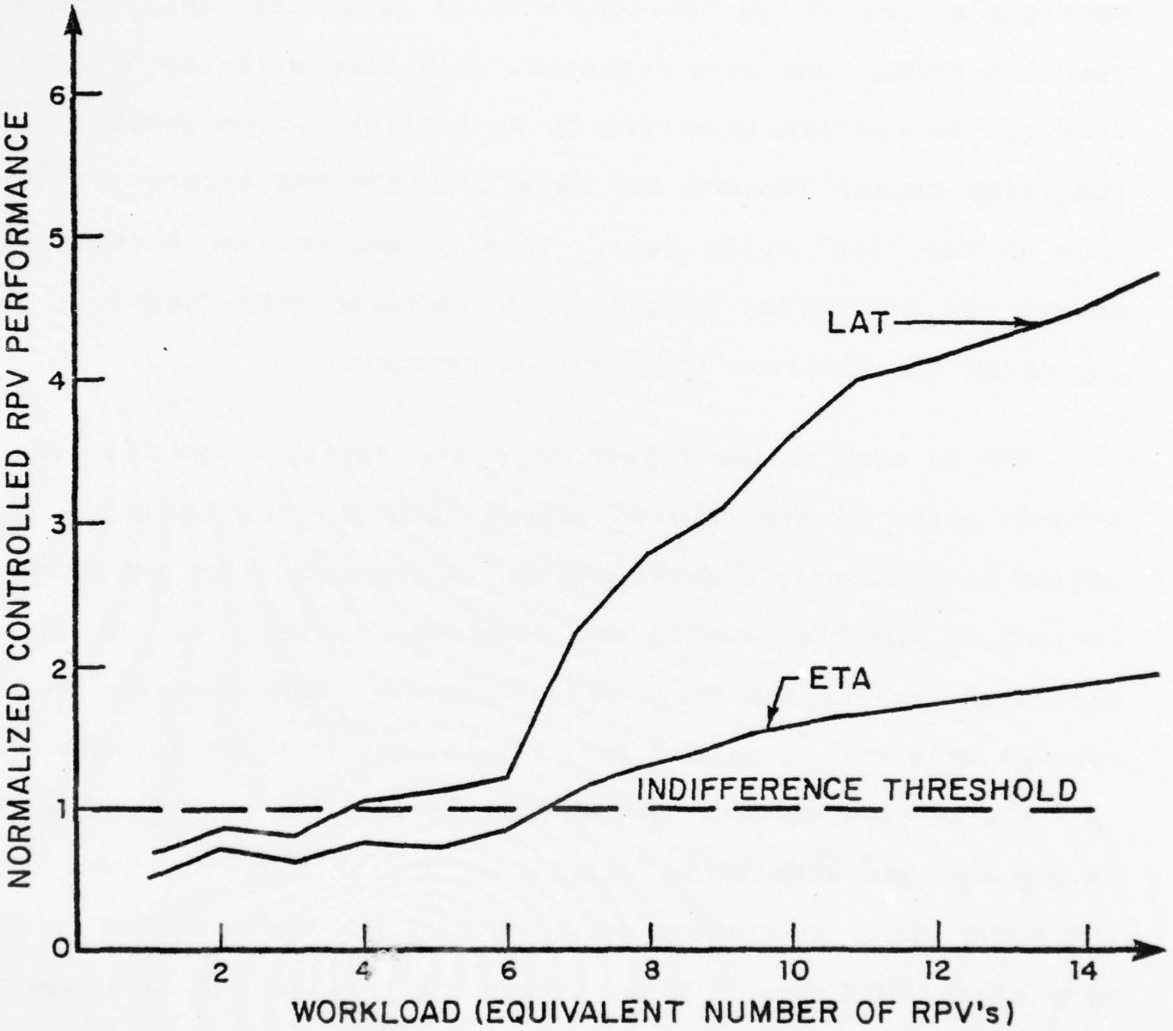


Fig. 14 Effect of Workload (Equivalent Number of Controlled RPVs) on Normalized Controlled Performance for High WØM.

operator has responsibility. We consider now the effect of the reporting error V_y and the navigation errors w_i on the system performance. Table 5 shows the effect of various levels of reporting errors V_y on the RMS deviations (averaged over the six controlled RPVs). It is clear that higher reporting errors lead to degradation in performance. The mean error in pop-up time is also shown. We would expect the errors in pop-up time to degrade steadily with increasing reporting errors if these errors were obtained by averaging over many runs (i.e., by Monte Carlo simulation).

Table 5: Effect of Reporting Errors

Vy	-20	-17	-14
RMS ETADEV (1000 ft.)	5.57	5.94	6.33
RMS ETADEV (1000 ft.)	0.95	1.02	1.15
MEAN ERROR in pop up time (sec.)	12.5	8	20

The effect of navigation errors is shown in Table 6 for three

Table 6: Effect of Navigation Errors

	CASE A	CASE B	CASE C
w_1 ft/sec	0.6	6.7	6.7
w_2 ft/sec	0.8	0.8	2.0
RMS ETA DEV (1000 ft.)	5.57	9.13	10.38
RMS LAT DEV (1000 ft.)	0.95	1.75	2.76
MEAN ERROR in pop up time (sec.)	12.5	24.2	22.5
NUMBER OF PATCHES (ETA + LAT DEV)	130	146	152

cases. Case A is the basic case, Case B has higher ETA navigation errors over Case A and Case C has higher LATDEV navigation errors over Case B. The results again show degradation in RMS deviations as the navigation errors get worse. Table 6 also shows that average number of patches per RPV also increases. The mean error in pop-up time is also displayed in Table 6 and the discussion in the previous paragraph applies here as well.

4.4 Remarks

In summary, the results obtained with the DEMON model are very interesting and are representative of the type of results that may be obtained with a top-down approach to modelling the RPV conrol problem. Of course, more sensitivity results would be useful but the results obtained thus far for monitoring performance show that the model does behave reasonbly, that the parameters do significantly affect the performance and that the monitoring and patching trends are as expected. They seem to be sufficient to capture the important aspects of variations in monitoring and patching strategies. They also show how the model may address important considerations from the system designer's point of view such as RPV/Operator ratio, allowable navigation errors, tolerable reporting errors and so forth. However, in the present version of DEMON details of control and display implementations are not so readily addressed, although these may be dealt with by further refinements of the model.

Thus, the results obtained do appear to provide a 'proof of concept' for the top-down approach employed in DEMON for modelling the RPV enroute control problem. However, validation of the model requires comparison of the results from DEMON with data from a 'real' RPV mission, which remains to be done.

5. CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

In this report, we have described the application of a decision-making, monitoring and control model (DEMON) of the human operator to a simplified RPV control task. However, our major objective in this effort was not so much to develop a model of RPV control as it was to explore issues in model development for complex systems and to compare this top-down approach with a bottom-up approach to the same problem. Unfortunately, a complete evaluation of the approaches cannot be made without model validation results and these are not yet available. Thus, we will confine our remarks here to a discussion of the DEMON approach to modelling complex systems in light of our experiences to date. The reader is referred to Reference 4 for an analysis of the bottom-up approach.

The DEMON model is aimed at addressing system level questions. It views the operator as an element in a closed-loop system whose decisions and actions are based on rational considerations concerning the task objectives. DEMON utilizes the information processing structure of the OCM which has been validated many times in the context of continuous control and in decision tasks.

A major question concerning the use of control theoretic techniques for modelling supervisory control tasks is the ability to treat asynchronous, discrete tasks in what is essentially a continuous set-up. Our results show that this is not a problem, conceptually at least, if one infuses decision theoretic notions into the model.

A question frequently raised with respect to the top-down modelling approach concerns the level of parametric specification needed for the model. This did not seem to be a major problem in this investigation. In the standard OCM, one would normally have to specify parameters that describe human limitations, namely, time delay and observation and motor noises. This was not necessary here because these limitations could be neglected in view of the long frame time and the reporting errors. We suspect that this might also be true for many other supervisory control tasks. Thus, we were left only with a parameter that described the operator's expectation as to the rate of growth of uncertainty due to navigation errors and a set of parameters associated with the expressions for expected net gain. These parameters relate directly to mission objectives and instructions or training. They are equivalent, in some sense, to a subset of the Operator Attribute file of the Pritsker bottom-up simulation. However, they are fewer in number and, more importantly, enter the model in quite a different way.

The manner in which decision making is implemented in DEMON leads to a significant difference between it and the bottom-up models of the RPV control task. In the bottom-up models, RPVs are looked at sequentially as determined from predetermined lists. In DEMON, RPVs are given synchronous consideration in that the operator is assumed to have an expectation with respect to the errors and relative importance of all RPVs at all times. An evaluation is then made as to which RPV and error has the highest priority (highest expected gain from monitoring) and that RPV is selected for observation.

The system approach used in DEMON avoids consideration of factors which have often been the traditional concern of human engineering. In the RPV control problem, we have not considered, at all, the motor aspects of the task. Nor have we focused on the questions of display configuration. Instead, we have concentrated on the cognitive portions of the problem (information processing and decision making). We believe this to be the major aspect of most supervisory control problems given the time constants usually involved in supervisory control. However, it is certainly true that errors that result from poorly designed operator/machine interfaces can be very significant and it would be desirable to have models capable of dealing with these issues.

To summarize, top-down models are generally appropriate for skilled operators who are attempting to act rationally to achieve well-defined objectives. They have been applied principally in the area of vehicle control, but DEMON provides a model structure with potential for application to problems in which human control actions are infrequent and in which monitoring and decision-making are the operator's main activities. DEMON contains within it the means for predicting the complex interactions among subtask components making up the task without specific enumeration of these interactions. The models for monitoring and decision-making are dynamic and are intrinsically adaptive, within given constraints, in response to the mission requirements as specified in the model's parameters. It is this property that provides the model with its potential predictive power. On the other hand, the top-down approach imposes a critical need for defining dimensions and/or parameters from which to formulate meaningful performance indices or decision criteria. In addition, there are major questions concerning the model's ability to deal with procedural aspects of human performance, to treat the problems that are the traditional concern of human factors and to handle multi-operator problems.

There are several areas where we believe further research is warranted. Some of these relate to modelling of the RPV control task and others are more general.

With respect to the RPV model, several simplifications were made to aid understanding and to allow us to consider a larger number of RPVs. It would be very worthwhile to modify the model so that it is both more realistic and more readily compared with the bottom-up models and the simulation data.* The first step in such a modification would be to go to a four state representation for RPV dynamics. Next, the display equations should be defined more realistically, i.e., so the information available from the various displays corresponds to that of the actual displays. Third, better models for navigation errors and command link status should be added. Fourth, the basic simulation time increment should be allowed to be less than a time frame so that decisions could be triggered out other than integral frame times. These changes are conceptually straightforward and relatively easy to implement; however, they will increase the computational costs.

During implementation of these changes further sensitivity analysis and model investigation is desirable. For example, we should explore the differences in performance that result from a strategy of considering ETA and LATDEV errors independently in deciding what to monitor (as was done here) as opposed to one of deciding which RPV to monitor (as is done in the bottom-up model).

* This would require enlarging our computer programs, as well, so that we could still consider a significant number of RPVs.

This should then be followed by a detailed comparison of model results with data and with results from other models.

Finally, two areas of general research interest are the extension of the modelling concepts to multi-operator situations and the development of integrated models that incorporate the desirable features of both the bottom-up and top-down approaches. Each of these areas poses problems of significant difficulty but the payoff from successful development would be considerable.

APPENDIX A

Bibliography of Reports and Papers and Presentations Produced Under Contract.

Pew, R. W., S. Baron, C. E. Feehrer and D. C. Miller, "Critical Review and Analysis of Performance Models Applicable to Man-Machine Systems Evaluation," BBN Report No. 3446, March 1977.

Miller, D. C., C. E. Feehrer, R. Muralidharan, R. W. Pew and S. Baron, "Development of Human Performance Models for Man-Machine Systems Simulation," BBN Report No. 3739, Oct. 1978.

Muralidharan, R. and S. Baron, "Combined Monitoring Decision and Control Model for the Human Operator in a Command and Control Task," Proceedings of the Fourteenth Annual Conference on Manual Control, NASA Conference Publication, 2Ø6Ø.

Pew, R. W. and S. Baron, "Man-Machine Performance Modeling," Presented at OSR Review of AF Sponsored Basic Research in Visual Processes and Human Operator Control, Dayton, Ohio, Sept. 1978.

Muralidharan, R. and S. Baron, "DEMON: A Human Operator Model for Decision Making, Monitoring and Control," To be presented at the Fifteenth Annual Conference on Manual Control, Dayton, Ohio, March 1979.

REFERENCES

1. Pew, R. W., S. Baron, C. E. Feehrer and D. C. Miller, "Critical Review and Analysis of Performance Models Applicable to Man-Machine Systems Evaluation," BBN Report No. 3446, March 1977.

2. Mills, R.,R. Bachert and N. Aume, "Supplementary Report of the RPV System Simulation Study II : Evaluation of RPV Position Report Smoothing and Automatic Heading Correction", AMRL-TR 75-87, September 1975.

3. Wortman, D., S. Duket and D. Seifert, "SAINT Simulation of a Remotely Piloted Vehcile/Drone Control Facility," Proceedings of 19th Annual Meeting of the Human Factors Society, 1975, 342-346.

4. Miller, D., Feehrer, C., Muralidharan, R., Pew, R. and Baron, S., "Development of Human Performance models for Man-Machine System Simulation", BBN Report No. 3739, October 1978.

5. Baron, S., "A Model for Human Control and Monitoring Based on Modern Control Theory," **Journal of Cybernetics and Information Sciences**, Vol. 4, No. 1, Spring 1976.

6. Carbonell, J. R.,"A Queueing Model of Many-Instrument Visual Sampling," **IEEE Transactions on Human Factors in Electronics**, Vol. HFE-7, No. 4, December 1966, pp. 157-164

7. Senders, J.,J. Elkind, M. Grignetti and R. Smallwood, "An Investigation of the Visual Sampling Behavior of Human Observers", NASA CR-434, April 1966

8. Kleinman, D. L., S. Baron and W. H. Levison, "An Optimal-Control Model of Human Response, Part 1: Theory and Validation," **Automatica**, Vol. 6, 1970, pp. 357-369